MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Viktor Bahýľ

Univerzální Turingův stroj

Katedra Algebry

Vedoucí bakalářské práce:	prof. RNDr. Jan Krajíček, DrSc.
Studijní program:	Matematika
Studijní obor:	Matematika pro informační technologie

Praha 2019

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 16. 5. 2019 ....

Podpis autora

Poďakovanie

Na tomto mieste by som sa rád poďakoval vedúcemu práce profesorovi RNDr. Janovi Krajíčkovi, DrSc., ktorý mi poskytol odborné rady a usmernenie pri písaní práce, spolu s informáciami, cennými radami a pripomienkami k danej téme, ktoré mi boli nápomocné pri tvorbe tejto práce.

Název práce: Univerzální Turingův stroj

Autor: Viktor Bahýľ

Katedra: Katedra Algebry

Vedoucí bakalářské práce: prof. RNDr. Jan Krajíček, DrSc., Katedra algebry

Abstrakt: Práce se zaměřuje na problematiku výpočetních úloh a jejich řešení. Na jejich řešení lze využít model Turingova stroje, který je plnou náplní této práce. Pro samotné řešení problémů je důležitá obecnost daného mechanismu, která je v teorií Turingova strojů zastoupena v podobě jejich univerzálnosti, která spojuje pojmy simulace a reprezentace. Hlavním výsledkem a úkolem práce je formulovat univerzální Turingův stroj a následně o něm tuto univerzálnost dokázat. V úvodních částech práce se nacházejí teoretické definice potřebné pro práci s těmito metodami řešení výpočetních úloh. Jako klíčové se ukáží již zmíněné pojmy simulace Turingových strojů a jejich reprezentace. Na oba pojmy je kladen samostatný důraz a snaha o co nejlepší vysvětlení těchto pojmů. V práci se pracuje s vlastní formou reprezentace, která je podrobně popsána a s nadefinovaným Turingovým strojem, jehož definice je také podrobně popsána, spolu s kompletním důkazem o univerzálnosti tohoto stroje.

Klíčová slova: Turing, stroj, univerzálnost, reprezentace

Title: Universal Turing machine

Author: Viktor Bahýľ

Department: Department of Algebra

Supervisor: RNDr. Jan Krajíček, DrSc., Department of Algebra

Abstract: This thesis is focused on the processes of solving computational problems. The Turing machine is an example of a model which we can use to solve these problems and these machines are the main objective of this Bachelor thesis. The generality of the model is important; it allows us to simulate any conceivable algorithm. In the theory of Turing machines, the generality is demonstrated by the construction of a universal Turing machine. This is the task of this Thesis: to define a universal Turing machine and to prove its universality. Key definitions, linked with the Turing machines, are recalled at the beginning of the Thesis. The simulation and representation of Turing machines will prove to be the key concepts. We make an extra effort to explain these fundamental notions. The thesis has its own form of representation and defined Turing machine with detailed descriptions for them, together with complete proof of the universality of the mentioned Turing machine.

Keywords: Turing, machine, universality, representation

Obsah

Úvod

V súčasnej dobe je rozvoj informačných technológií jedným z najviac sa rozvíjajúcich odvetí. Samotné informačné technológie sa budovali mnoho rokov a k ich vytvoreniu viedla všeobecná snaha vytvoriť jednotný model, ktorý by poskytoval riešenie pre najrôznejšie výpočtové úlohy, z čoho pochádza aj názov týchto technológií v tomto odvetví, ktoré často označujeme ako výpočtové.

O skonštruovanie a zavedenie takýchto modelov, ktoré by dokázali všeobecne formulovať riešenia prebehlo mnoho pokusov. O zavedenie takéhoto pojmu, teda všeobecnej metódy, sa pokúsil aj Alan Turing, britský matematik, podľa ktorého boli pomenované takzvané Turingové stroje. O týchto strojoch pojednáva táto bakalárska práca, ktorá poskytuje všeobecný náhľad nato, ako Turingov stroj funguje. Pričom sa zameriava na zložitejšie výsledky tejto teórie, ako je napríklad existencia špeciálneho Turingovho stroja, nazývaného univerzálny Turingov stroj.

V práci sa postupne nachádzajú formálne definície jednotlivých pojmov spolu s vysvetlením a priblížením pozadia za týmito pojmami. Postupne pokladáme teóriu na zavedenie a ukážku reprezentácie jednotlivých strojov, ktorá sa ukáže ako kľúčový výsledok k schopnosti uvažovania nejakého všeobecného algoritmu v podobe univerzálnosti samotného výpočtu. Následne tieto poznatky zužitkujeme pri ukážke Turingovho stroja, o ktorom si zároveň dokážeme, že sa jedná o univerzálny Turingov stroj s predpokladmi, ktoré môžeme považovať v tejto teórií ako za rozumné.

1 Definícia Turingovho stroja

Základom tejto práce je pojem Turingov stroj. Pre pochopenie tohto pojmu sa najskôr pozrieme na intuitívny náhľad k problematike výpočtových úloh, ktoré sú súčasťou nie len samotnej matematiky, ale aj iných vedeckých či nevedeckých odvetví.

Základné ponímanie, ktoré viedlo aj k prvotnému definovaniu Turingovho stroja, je riešenie takýchto výpočtových úloh. Pre ich riešenie sa ľudia dlhodobo snažili nájsť nejaký všeobecný matematický model, ktorý by dokázal pokryť ak nie všetky, tak aspoň väčšinu možných algoritmov pre riešenie daných problémov a dospieť tak k uspokojivému výsledku vo väčšine prípadov alebo inak povedané pre čo najväčšiu množinu vstupov.

Na prvý pohľad sa zdá, že takýto model nemôže existovať, našťastie sa neskôr ukázalo, že takýto model predsa len existuje a jedná sa o už skorej spomínaný Turingov stroj. Pred samotnou formálnou definíciou tohto pojmu sa najskôr pozrime na jeho jednoduchú interpretáciu.

1.1 Jednoduchá interpretácia Turingovho stroja

Na výpočtové úlohy hľadíme ako na aplikáciu mechanických pravidiel v procese manipulácie s číslami. Ten čo počíta, môže sa jednať o osobu riešiacu daný problém, prípadne o stroj, ktorý by daný proces vedel automatizovať, má k dispozícií zápisník, na ktorý si môže zapisovať medzivýsledky v priebehu výpočtu. Tento pohľad využíva aj nami pozorovaný pojem Turingov stroj.

Predpokladajme, že problém, ktorý chceme riešiť sa dá zapísať pomocou jednej funkcie s binárnym reťazcom ako výstupom. Takúto funkciu si označme f. Vstupom do funkcie bude reťazec bitov a algoritmus, ktorý danú funkciu f spočíta, je nejaká konečná množina mechanických pravidiel. Tieto pravidlá nemusia spĺňať to, že pre každý možný vstup vrátia hodnotu $f(x)$, kde x značí bitový reťazec, ale musia spĺňať, že pre jeden vstup vrátia najviac jeden výstup, prípadne nedôjde k ukončeniu výpočtu, ale ak zopakujeme výpočet pre rovnaký vstup vždy nastane rovnaká situácia. Avšak táto množina pravidiel je pre každý problém rovnaká a nemenná. Pričom jednotlivé pravidlá môžu byť aplikované viackrát a pozostávajú z jednej alebo viacerých nasledujúcich elementárnych operácií:

1. Prečítaj bit zo vstupu.
2. Prečítaj bit (alebo symbol z väčšej abecedy, povedzme číslicu z {0, ... , 9}) zo zápisníka alebo pracovného priestoru, ktorý daný algoritmus môže využívať.

   Na základe prečítaných hodnôt,

1. Zapíš bit alebo symbol do zápisníka.
2. Buď ukonči algoritmus s výstupom, prípadne zvoľ nové pravidlo zo zoznamu, ktoré sa bude vykonávať ako ďalšie.

1.2 Formálna definícia

V nasledujúcej sekcii sa nachádza samotná formálna definícia pojmu Turingovho stroja, spolu s niektorými základnými faktami o tomto pojme. Pre potreby tejto práce definujeme Turingov stroj nasledovne podľa formálnej definície a budeme ju využívať naprieč celou prácou.

Definícia 1 (Turingov stroj):

Turingov stroj M definujeme ako usporiadanú množinu obsahujúcu sedem prvkov $M = (k, \Gamma, \square, V, Q, q_0, \delta)$, kde $k \in \mathbb{N}$, platí $k \geq 2$ a k označuje počet pások stroja M. Γ, V a Q sú konečné množiny. Funkcia $\delta: Q \times \Gamma^k \rightarrow Q \times \Gamma^{k-1} \times \{L, S, R\}^k$ je prechodová funkcia, kde množina {L, S, R} označuje smer pohybu hláv. Uvažujeme aj čiastočne definovanú prechodovú funkciu. Konečnú množinu Γ nazývame abecedou stroja M, $\square \in \Gamma$ je prázdny symbol, označujúci prázdne políčko na páske. Konečná množina V sa nazýva vstupno-výstupnou abecedou stroja M a platí $V \subseteq \Gamma \setminus \{\square\}$, okrem toho predpokladáme, že platí $\{0, 1\} \subseteq V$. Konečná množina Q obsahuje všetky stavy, v ktorých sa stroj M môže nachádzať a $q_0 \in Q$ je štartovací stav.

V definícii môžeme pracovať s viacerými obmenami a zmenami bez toho, aby sme prišli o správnosť chodu stroja, alebo zmenšili silu tohto matematického nástroja. Táto vlastnosť svedčí o robustnosti samotnej definície a poskytuje nám zaujímavý pohľad do danej problematiky. V stroji pracujeme so všeobecným prirodzeným číslom *k*, ktoré nám udáva počet pások, ako však uvádza Sipser (2013) vo Vete 3.13, ukazuje sa, že *k*-páskový Turingov stroj má podobné, ak nie rovnaké vlastnosti, ako keby sme pracovali len s pevným *k*, napríklad jedno-páskovými strojmi. Túto vlastnosť neskôr využijeme pri samotnom návrhu univerzálneho Turingovho stroja. Bolo by teda možné pracovať len s jednopáskovým strojom, ale pre prehľadnosť práce a jednoduchší náhľad do problematiky budeme pracovať so všeobecným k. Ukážeme ako by daný stroj pracoval v takejto podobe s tým, že dané výsledky sa dajú previesť aj na jedno-páskový stroj M, ktorý by sme museli špeciálne definovať.

Z našej definície je však jasné, že stroj využíva minimálne dve pásky, pričom všeobecne označujeme toto číslo k. Samotnou páskou nazývame nekonečnú polpriamku pozostávajúcu z pamäťových blokov, čo znamená, že po konečnom počte krokov sa stroj dostane na jej ľavý okraj, pričom do pravej strany môže teoreticky pokračovať do nekonečna. Na týchto páskach má stroj M zapísaný vstup, v priebehu výpočtu si na tieto pásky zaznamenáva medzivýsledky a na záver na tieto pásky zapíše aj výstup.

Na každej z týchto pások sú v jej pamäťových blokoch zapísané symboly z abecedy stroja M. Po jednotlivých blokoch sa pohybujeme pomocou k páskových hláv, pre každú pásku máme aj jednu jej hlavu a teda ich označujeme rovnako ako pásky. Preto niekedy nemusíme špecifikovať, či hovoríme o páske alebo jej hlave, pričom z kontextu by malo byť jasné, o ktorej časti stroja hovoríme. Tieto hlavy teda ukazujú na konkrétny blok na ich páskach a práve s týmto blokom, v jednotlivých krokoch chodu stroja M, vieme pracovať. Tieto hlavy vedia z daného bloku čítať, zapisovať nový symbol z abecedy stroja M a posunúť sa o jeden blok, jedným alebo druhým smerom, prípadne zostať na mieste. Pokiaľ sa hlava nachádza na najľavejšom konci jej pásky a pokúsi sa pohnúť viac doľava, tak namiesto toho zostane na mieste.

Vstup do Turingovho stroja M, tvorí reťazec nad jeho vstupno-výstupnou abecedou V. Teda nech $x \in V^*$, potom povieme, že vstupom do stroja M je x pomocou nasledujúceho zápisu $M(x)$. Tento vstup si potrebujeme zapísať na samotnú pásku stroja M.

Na tento účel si vyčleníme prvú z pások, ktorú budeme označovať ako vstupnú pásku stroja M. Na tejto vstupnej páske sa bude nachádzať vstup stroja M a to tak, že vstupný reťazec x zapíšeme na prvú pásku stroja na jej začiatok. Z tejto pásky vieme len čítať a nedokážeme do nej zapisovať, aby sme nepozmenili vstup počas chodu stroja M, čo znamená, že vstupná hlava nedokáže zapisovať nový symbol dokáže len čítať.

Zvyšné pásky označíme ako pracovné a poslednú z nich zase ako výstupnú pásku stroja *M*. Keďže stroj môže mať potenciálne len jednu pásku určenú aj ako pracovnú a aj výstupnú, tak potom tieto pásky splývajú, preto predpokladáme, že výstup chodu stroja sa vždy nachádza na začiatku poslednej pásky stroja *M*, hneď na prvej pozícií a ukončený je vždy prázdnym symbolom. Na pracovné pásky si stroj zaznamenáva medzivýsledky počas výpočtu funkcie, ktorú reprezentuje. V prípade stroja kde splývajú pracovná a výstupná páska stroja si tieto medzivýsledky zapisuje na poslednú pásku a výstupom potom prepíše prípadne záznamy nachádzajúce sa na jej začiatku. Môžeme teda využiť názov pracovnej pásky len pre také pásky, kde sa nenachádza výstup a sú využívané len na zaznamenávanie medzivýsledkov počas výpočtu. Pojem výstupná páska potom bude vždy oddelený od pracovných pások, ako posledná páska stroja *M* obsahujúca výstup stroja. Pričom pri niektorých strojoch sa jedná o tú istú pásku, ale počas výpočtu ju nazývame pracovnou a až po zapísaní výstupu a ukončení chodu stroja ju nazveme výstupnou. Môžeme teda formálne nadefinovať výstup Turingovho stroja.

Definícia 2 (Výstup Turingovho stroja):

Nech $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ je Turingov stroj. Pre vstup Turingovho stroja M, pri ktorom dôjde k ukončeniu chodu definujeme jeho výstup ako najdlhší súvislí reťazec znakov vstupno-výstupnej abecedy V, nachádzajúci sa na jeho k-tej páske začínajúci na prvom poli a ukončený prázdnym symbolom □. Pokiaľ je na prvej pozícií uložený prázdny symbol, výstupom je prázdny reťazec. Výstup stroja M označíme M(x), kde $x \in V^*$ je vstup. Pokiaľ k ukončeniu chodu na vstupe Turingovho stroja nedôjde, nedefinujeme jeho výstup.

Množina *Q* obsahuje stavy stroja *M*, v ktorých sa môže tento stroj nachádzať. Tieto stavy sú potom jedným zo vstupov do prechodovej funkcie. Na začiatku sa stroj *M* nachádza v štartovacom stave $q_0 \in Q$, ktorý ešte pred vykonávaním chodu predpokladá, že stroj je v takzvanej štartovacej konfigurácií. Teda na každej páske sa vo všetkých blokoch nachádzajú zapísané prázdne symboly, s výnimkou vstupnej pásky, kde je na konečnom počte prvých blokov zapísaný vstup stroja *M*. Ďalej sú hlavy stroja *M* umiestnené na začiatku každej pásky, čo znamená, že sú na najľavejšom okraji každej z nich a stroj sa nachádza v štartovacom stave q_0 z množiny Q.

V sekcii 1.1 sme predpokladali, že funkcia f má pre každý vstup nanajvýš jeden výstup. Keďže náš pohľad na model Turingovho stroja spočíva v reprezentácií takejto funkcie hovoríme o deterministickom Turingovom stroji, pre ktorý platí, že jeho prechodová funkcia má pre každý vstup, pre ktorý je definovaná len jeden výstup. Existujú aj nederministické stroje, pre ktoré platí, že ich prechodová funkcia má rozhodovací charakter a pri jednotlivých vstupoch môže dochádzať k voľbe výstupu. Sipser (2013) vo Vete 3.16 na str. 178 dokázal, že pre každý nedeterministický stroj existuje k nemu deterministický, ktorý ho nahrádza, pričom v našej terminológií neskôr povieme, že ho simuluje (viď Definícia 3). Preto sa týmito strojmi nemusíme v práci zaoberať.

Prechodová funkcia určuje samotný chod stroja M. Jedná sa o algoritmus, ktorý stroj dodržiava a nasleduje, pričom jeho chod nemusí nikdy skončiť, čo znamená, že vždy existuje ďalší predpis pre prechodovú funkciu, hovoríme teda, že stroj pre konkrétny vstup neskončí. Zaujímavejšie je však pozorovať také stroje, ktoré skončia, to znamená, že po konečnom počte krokov neexistuje ďalšia inštrukcia pre danú prechodovú funkciu a v takomto prípade sme definovali výstup Turingovho stroja. K ukončeniu chodu stroja dôjde v momente, keď nemá prechodová funkcia stroja definovanú ďalšiu vykonateľnú inštrukciu, čo znamená, že je to čiastočne definovaná funkcia, teda pre niektorý vstup do nej neexistuje výstup a v takomto momente hovoríme, že dôjde k ukončeniu chodu stroja M.

Počas vykonávania prechodovej funkcie stroj číta a zapisuje pomocou hláv symboly zo svojej abecedy na svoje pásky, posúva hlavy požadovaným smerom a tiež prechádza množinu stavov Q. Vstup do prechodovej funkcie predstavuje aktuálny stav stroja q a k-tica prvkov abecedy stroja M, pozostávajúca z prvkov načítaných z k hláv stroja M. Zatiaľ čo výstup prechodovej funkcie pozostáva z nového stavu z množiny Q, ktorý nahradí aktuálny, ďalej z k-1-tice prvkov abecedy stroja M, ktorá sa zapíše na k-1 posledných pások. Poslednou časťou výstupu je k-tica prvkov nad množinou $\{L, S, R\}$, kde symbol L reprezentuje posunutie hlavy doľava, S ponechanie hlavy na mieste a R posunutie hlavy doprava. Pri symbole L je potrebné dodať, že ak sa hlava už nachádza na najľavejšom okraji pásky tak ponecháme túto hlavu na mieste.

1.3 Definícia univerzálneho Turingovho stroja

Pre definíciu univerzálneho Turingovho stroja si najskôr potrebujeme zadefinovať niekoľko pojmov, ktoré pri jeho definícií využijeme a budeme s nimi pracovať aj v ďalších častiach práce.

Najskôr sa pozrieme na vzťah prechodovej funkcie a stroja M. Každý stroj takúto funkciu obsahuje. Keď sa zamyslíme nad samotnou definíciou Turingovho stroja zo sekcie 1.2, uvidíme, že táto funkcia dokáže samostatne popísať celé fungovanie a chod ľubovoľného Turingovho stroja M. Pokiaľ by sme teda dokázali nájsť spôsob, ako túto funkciu reprezentovať v tvare reťazca znakov nejakej abecedy, ktorá bude obsahovať abecedu stroja M, vedeli by sme takýto reťazec predať ako vstup do iného Turingovho stroja. V takomto popise by sme vedeli nájsť všetky stavy stroja M, vedeli by sme si dohľadať počet pások stroja a vedeli by sme zrekonštruovať samotnú modulačnú funkciu stroja M. Keďže dokážeme slovne popísať prechodovú funkciu, tak je možné pomocou kódovania previesť symboly anglickej abecedy na ľubovoľné symboly, pomocou ASCII kódu a teda dokážeme takýto popis vždy vytvoriť pre ľubovoľný stroj nad požadovanou abecedou. Nazvime teda takýto popis zatiaľ neformálne ako *reprezentáciou Turingovho stroja M*, pričom formálne si nejakú reprezentáciu nadefinujeme neskôr v sekcii 2.1, nateraz nám teda stačí vedieť, že nejaká existuje.

Pre každý stroj M budeme používať značenie m pre nejakú jeho reprezentáciu. Reprezentácia nie je jednoznačná a každý stroj môže mať viacero reprezentácií. Konkrétny tvar reprezentácie, ktorú budeme ďalej využívať si konkrétne zadefinujeme v už spomínanej sekcii 2.1.

Niektoré Turingové stroje môžu mať vlastnosť, ktorej hovoríme, že jeden stroj simuluje druhý. To znamená, že jeden stroj dokáže riešiť problémy druhého stroja a dôjsť k rovnakému výstupu, pričom vstupno-výstupná abeceda stroja, ktorý simuluje, obsahuje vstupno-výstupnú abecedu simulovaného stroja. Takýto vzťah medzi strojmi by sme si radi definovali a pracovali s ním v ďalšom priebehu práce. Definujme teda tento pojem nasledovne.

Definícia 3 (Simulácia Turingových strojov):

Nech M a R sú Turingové stroje, označme ich vstupno-výstupné abecedy v poradí V_M *a* V_R*. Potom povieme, že Turingov stroj M simuluje R, pokiaľ* $V_R \subseteq V_M$*, a pre všetky* $x \in V_R^*$ *platí:*

(a) *M ukončí pre vstup x svoj chod* $\Leftrightarrow$ *R ukončí pre vstup x svoj chod,*
(b) *Ak je splnené (a) potom sa výstupy M a R v x rovnajú, píšeme* $M(x) = R(x)$*.*

Ako vidíme bod (a) hovorí, že stroje musia ukončiť svoj chod v rovnakých prípadoch. Túto podmienku vyžadujeme preto, že výstup stroja sme definovali len v prípade kedy dôjde k ukončeniu jeho chodu. Z definície vyplýva, že nám nezáleží na samotných definíciách strojov, zaujíma nás len či dané stroje dodajú rovnaké výsledky a teda či reprezentujú rovnaký výpočtový problém.

Teraz môžeme pristúpiť k zadefinovaniu univerzálneho Turingovho stroja, pričom budeme pracovať s pojmom reprezentácie Turingovho stroja. Keďže bod (b) v nasledujúcej definícií je iba preformulovanie pojmu simulácie, budeme môcť hovoriť, že univerzálny Turingov stroj *U* simuluje stroj *M*.

Definícia 4 (Univerzálny Turingov stroj):

Povieme, že U je univerzálny Turingov stroj pokiaľ

(a) $U = (k_u, \Gamma_u, \square, V_u, Q_u, q_{0u}, \delta_u)$ *je Turingov stroj taký, že* $0, 1 \in V_u$.
(b) *Pre každý Turingov stroj* $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ *platí, že* $V \subseteq V_u$ *a pre každé* $x \in V^*$ platí, že *M ukončí svoj chod na vstupe x* práve vtedy keď *U ukončí svoj chod na vstupe (x, m) a keď M ukončí svoj chod na vstupe x, potom platí rovnosť* $M(x) = U(x, m)$*, kde* $m \in V_u^*$ *je nejaká reprezentácia stroja M.*

Vidíme, že vstupom do univerzálneho Turingovho stroja *U* môže byť ľubovoľný stroj *M*, s čím musíme rátať a preto sa nám zídu predpoklady nadefinované na začiatku kapitoly 2. Tieto predpoklady využijeme pri návrhu nášho univerzálneho stroja, v sekcii 2.2. Vidíme, že veľmi dôležitým prvkom pre univerzálny stroj je zvolenie vhodnej reprezentácie jednotlivých strojov, ktoré budeme chcieť simulovať. V definícii uvádzame, že vstup do univerzálneho stroja je tvorený dvoma časťami, prvou časťou je už spomínaný vstup, ktorý by sme inak zadávali do stroja *M*, ktorý chceme simulovať a pre ktorý hľadáme výstup. Pričom druhú časť, tvorí reprezentácia Turingovho stroja *M*. Viac zložkový vstup si konkrétnejšie opíšeme v sekcii 2.2.

2 Univerzálny Turingov stroj

V tejto kapitole pristúpime k samotnej definícií univerzálneho Turingovho stroja U. Pre začiatok však nahliadnime niekoľko faktov, ktoré budeme brať ako všeobecné predpoklady pre Turingové stroje. Bolo by možné tieto predpoklady vynechať a pracovať bez nich, ale v tejto práci sa chceme skôr zamerať na prehľadnosť fungovania tohto pojmu ako takom a nie úplne na efektívnosť danej simulácie Turingovho stroja.

V prvom rade by sa nám hodil predpoklad, že Turingov stroj, ktorý budeme simulovať, bude pracovať s najjednoduchšou možnou vstupno-výstupnou abecedou obsahujúcou len binárne hodnoty 0 a 1. Označme si túto úvahu do nasledujúceho predpokladu:

Predpoklad 1 (Zjednodušenie abecedy):

Nech $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ *je Turingov stroj, potom k nemu existuje stroj* $\tilde{M}$*, ktorý ho simuluje v tvare* $\tilde{M} = (k, \tilde{\Gamma}, \square, \tilde{V}, \tilde{Q}, \tilde{q}_0, \tilde{\delta})$*, kde* $\tilde{V} = \{0, 1\}$ *a* $\tilde{\Gamma} = \tilde{V} \cup \{\square\}$.

Tento predpoklad vyplýva z poznatkov, sformulovaných v (Arora a iný, 2009), konkrétne z Tvrdenia 1.5 na strane 16, ktoré hovorí o tom, že takáto simulácia je možná pokiaľ si symboly z väčšej abecedy zakódujeme pomocou binárnych symbolov 0 a 1 napríklad pomocou ASCII kódu. Samozrejme bude potrebné rozšíriť množinu stavov Q a upraviť prechodovú funkciu, ale daná simulácia je možná bez zmeny počtu pások, preto toto tvrdenie formulujeme ako predpoklad a nebudeme ho ďalej dokazovať.

Na základe tohto predpokladu môžeme teda uvažovať, že každý stroj M, ktorý budeme chcieť simulovať univerzálnym Turingovým strojom U bude používať abecedu $\Gamma = \{\square, 0, 1\}$ a vstupno-výstupnú abecedu $V = \{0, 1\}$.

V podobnom duchu by sme radi nahliadli, že každý stroj, ktorý budeme simulovať bude mať nejaký konštantný počet pások. Hodilo by sa nám, aby sme mohli o stroji M povedať, že bude mať tri pásky, z toho jednu vstupnú, jednu výstupnú a jednu pracovnú. Definícia 4 na str. 9 však hovorí, že U nazveme univerzálnym Turingovým strojom vtedy, keď simuluje ľubovoľný Turingov stroj M, ktorý nemusí mať pevný počet pások. Z Tvrdenia 1.6, na strane 17, zo zdroja (Arora a iný, 2009) vyplýva, že existuje zakódovanie hodnôt zo všeobecného počtu pások k na jednu jedinú pásku. Idea spočíva v tom, že jednotlivé údaje z k pások zapíšeme na $jk+i$-tú pozíciu na jednej páske, kde i je poradové číslo pásky a j je pozícia symbolu na i-tej páske v stroji M. Toto, teda bez dôkazu, sformulujme do nasledujúceho predpokladu:

Predpoklad 2 (Jedno-pásková simulácia):

Nech $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ je Turingov stroj, potom k nemu existuje Turingov stroj $\tilde{M} = (\tilde{k}, \Gamma, \square, V, \tilde{Q}, \tilde{q}_0, \tilde{\delta})$, kde $\tilde{k} = 3$, ktorý ho simuluje.

Výsledkom tohto predpokladu je, že náš univerzálny Turingov stroj U bude teda, bez ďalšieho odôvodňovania, vedieť kódovať v zmysle opísanom pred predpokladom a teda bude vedieť simulovať všetky pracovné pásky stroja M na jednu jedinú pracovnú pásku. Takže budeme môcť uvažovať len stroje M, pre ktoré platí, že ich k je rovné trom.

Spojením týchto dvoch predpokladov dostávame, že každý stroj M, ktorý budeme pomocou jeho príslušnej reprezentácie predávať Turingovmu stroju U, bude mať parametre $(3, \Gamma = \{\square, 0, 1\}, \square, V = \{0, 1\}, Q, q_0, \delta)$, kde Q, δ a q_0 sú ľubovoľné.

2.1 Reprezentácia Turingovho stroja

Ako sme si už povedali v sekcii 1.3 môže existovať niekoľko možných reprezentácií Turingových strojov. Jedna z takýchto reprezentácií bude náplňou tejto sekcie, spolu s konkrétnym popisom ako takáto reprezentácia bude vyzerať. Pričom na konci sekcie si ukážeme, že nami vytvorená reprezentácia skutočne jednoznačne popisuje jednotlivé Turingové stroje.

Našim hlavným bodom záujmu v tejto sekcii bude prechodová funkcia stroja M. Pripomeňme si teda jej definíciu. Prechodová funkcia Turingovho stroja $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ je definovaná nasledovne $\delta\colon Q \times \Gamma^k \rightarrow Q \times \Gamma^{k-1} \times \{L, S, R\}^k$. Vstupom sú teda ako aj všetky možné stavy stroja M tak aj k-tice prvkov z jeho abecedy. Usporiadajme si množinu Q, tak aby bol prvý stav štartovací a ostatné povedzme pomocou nejakého lexikografického zobrazenia, na samotnom usporiadaní nám nezáleží. Označme si teda jednotlivé stavy z množiny všetkých stavov Q nasledovne; prvý stav označíme q_0, druhý stav q_1 a tak ďalej až posledný stav označíme q_{n-1}, kde $n \in \mathbb{N}$ označujúce veľkosť množiny Q. Takto označené stavy potom môžeme kódovať pomocou binárnej sústavy nasledovným spôsobom. Najskôr si nájdeme najmenšie l spĺňajúce, že $2^l \geq n$. Stav q_i potom budeme kódovať na l pamäťových blokoch a to tak, že prevedieme číslo i do binárnej sústavy a doplníme prípadnými nulami na začiatok, aby v takomto ponímaní mal tento reťazec potrebnú dĺžku l. V rámci jedného stroja bude toto l vždy pevné a nemenné a pri predávaní reprezentácie môžeme uvažovať, že každý stroj môže mať toto l rôzne, nehrá to pre nás úlohu v tom zmysle, že nás pri simulácií univerzálnym strojom presná hodnota l nebude zaujímať, prípadne by sme si ju vedeli určiť zo samotnej reprezentácie. Jednoducho budeme len porovnávať hodnoty v dvoch pamäťových blokoch až pokiaľ nenarazíme na rozdielne symboly alebo na nejaký nami zvolený symbol označujúci koniec kódu stavu, aby sme vedeli určiť konkrétny stav, preto má zmysel uvažovať o takomto všeobecnom l.

Pre ilustráciu nech máme napríklad stroj M, pre ktorý platí, že $|Q| = 7$. Vidíme, že najmenšie l, spĺňajúce $2^l \geq 7$, je $l = 3$, takže budeme kódovať na troch bitoch. Nech $Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$, a teda v zmysle vyššie opísaného kódovania by sme dostali $Q = \{000, 001, 010, 011, 100, 101, 110\}$. Nazvime teda takéto kódovanie *označenie stavov v binárnej sústave* alebo *binárne označenie stavov*, prípadne *binárna reprezentácia stavov*, ako vidíme na l nám nezáleží a nemusíme ho uvádzať ani v názve. Pre každý stroj sa dá jednoducho dopočítať a univerzálny stroj ho nemusí poznať.

Ďalším vstupom prechodovej funkcie sú znaky abecedy stroja M načítané z k pások. Takže sa jedná o usporiadanú k-ticu nad Γ. V tomto momente môžeme využiť Predpoklad 2, ktorý sme si interpretovali tak, že budeme predpokladať to, že stroj M bude mať $k = 3$. Tieto trojice teda budeme reprezentovať tromi symbolmi bezprostredne nadväzujúcimi po sebe v takom zmysle, že prvý symbol v reťazci bude reprezentovať symbol z prvej pásky, druhý z druhej a tretí z tretej. Ako takýto reťazec pripojíme k binárnemu označeniu stavov? Pre túto úlohu si do abecedy zadefinujeme dva špeciálne znaky, ktoré budeme využívať len v rámci tejto reprezentácie. Predpoklad 1 hovorí, že stroj M má abecedu definovanú ako $\Gamma = \{\square, 0, 1\}$. Takže z tohto predpokladu vidíme, že žiadny stroj nebude využívať symbol / a ani @, čo je pre nás dôležité. Symbol @ využijeme pre označenie začiatku vstupu do prechodovej funkcie a teda sa bude nachádzať pred binárnym označením stavov. Za toto označenie následne pripojme znak / a za ním nech ďalej pokračuje reťazec znakov načítaných z trojice pások.

Pre každý stav však môže existovať viacero možných načítaných trojíc, v ktorých sa stroj môže ocitnúť. Takže by nemalo zmysel tento reťazec predlžovať a pred každou takouto trojicou písať aj označenie stavu. Preto tento reťazec začneme vždy @ s označením prvého stavu, následne spolu s / pripojíme všetky možné trojice, v ktorých sa môže tento stav nachádzať. Uvažujeme len tie trojice, ktoré pre daný stav majú zmysel a teda keďže prechodová funkcia môže byť čiastočná, nepíšeme všetky možne trojice nad abecedou Γ, ale len také, pre ktoré máme definované výstupy v opise stroja M. Následne tento výpis ukončíme symbolom @ a ak má stroj viac stavov, pokračujeme ďalším stavom v poradí a tak ďalej pokiaľ všetky stavy nepoužijeme. Reťazec vždy ukončíme symbolom @. Pokiaľ stav nemá žiadne vstupné trojice, tak za označením tohto stavu zapíšeme len symbol @, aby sme dali najavo, že ak je stroj v tomto stave má ukončiť svoj chod. Pre príklad nech $M = (3, \{\square, 0, 1\}, \square, \{0, 1\}, \{q_0, q_1, q_2\}, q_0, \delta)$ je Turingov stroj s prechodovou funkciou definovanou nasledovne:

$$\delta(q_0, 0\square\square) = (q_2, \square\square, \mathrm{SSS}),$$

$$\delta(q_0, 1\square\square) = (q_2, \square\square, \mathrm{SSS}),$$

$$\delta(q_2, 0\square\square) = (q_1, \square 0, \mathrm{RSS}).$$

Potom reťazec

@00/0□□/1□□@01@10/0□□@

reprezentuje vstupy do prechodovej funkcie stroja M, pričom M má tri stavy. V prvom má prechodová funkcia definované dve možné trojice, v treťom len jednu, zatiaľ čo v druhom nemá definovaný žiadnu možnú trojicu, takže keď sa stroj dostane do tohto stavu určite ukončí svoj chod. Podobne ak by sa stroj ocitol napríklad v stave q_2 s načítanou trojicou symbolov napríklad 1□□, tak vidíme, že aj v takomto prípade by stroj ukončil svoj chod. Zatiaľ sme žiadnym spôsobom nedefinovali výstup prechodovej funkcie v takomto reťazci, nato sa zameriame práve teraz.

Výstup z prechodovej funkcie tvorí trojica údajov. V definícií je ako prvý uvedený údaj o novom stave stroja, do ktorého sa má po vykonaní aktuálneho kroku prepnúť. Druhá je v našom prípade dvojica znakov nad abecedou Γ, ktorú zapíšeme na posledné dve pásky. A tretiu tvorí v našom prípade trojica informácií o tom, kam sa majú hlavy na jednotlivých páskach posunúť. Pre účely vykonávania prechodovej funkcie, sa nám bude hodiť túto trojicu informácií usporiadať do iného poradia a to tak, že začneme dvojicou zapisujúcich symbolov, nasledovať bude trojica informácií o posune a zakončíme to novým stavom stroja M. To znamená, že v našom príklade prepíšeme definíciu prechodovej funkcie nasledovne:

$$\delta(q_0, 0\square\square) = (\square\square, \mathrm{SSS}, q_2),$$

$$\delta(q_0, 1\square\square) = (\square\square, \mathrm{SSS}, q_2),$$

$$\delta(q_2, 0\square\square) = (\square 0, \mathrm{RSS}, q_1).$$

Pre každý vstup teda potrebujeme nadefinovať reťazec jeho výstupu z prechodovej funkcie. Medzi jednotlivými prvkami výstupu nebudeme pridávať žiadny symbol a nebudeme ani oddeľovať samotný výstup od vstupu. Jednotlivé symboly teda budú priamo nadväzovať na seba a budú vytvárať jednotný reťazec. Prvé dva symboly v tomto reťazci budú odpovedať novým symbolom, ktoré chceme zapísať na dané pásky, prvý z nich na druhú pásku a druhý z nich na tretiu (tento princíp odpovedá zápisu trojice načítaných symbolov pri vstupe, s rozdielom, že pracujeme len s dvojicou).

Nasledovať budú tri symboly, reprezentujúce kam sa má konkrétna hlava posunúť respektíve zostať na mieste. V definícií prechodovej funkcie je táto množina popísaná nasledovne $\{L, S, R\}$, kde L značí posun hlavy doľava, S zotrvanie hlavy na rovnakej pozícií a R zase posun hlavy doprava. My si túto množinu predefinujeme nasledovne $\{1, 0, \square\}$, kde 1 bude zastupovať L, 0 bude nahrádzať S a $\square$ bude reprezentovať R v našom reťazci. Tieto symboly budú vždy v trojici a prvý z nich bude udávať, čo sa má stať s prvou hlavou, druhý zase s druhou hlavou a tretí s tou treťou.

Reťazec výstupu bude uzatvárať nový stav, ktorý sa má zapísať ako nový aktuálny stav q. Tento nový stav budeme rovnako ako pri vstupe reprezentovať binárnym označením stavov, s tým rozdielom, že ho pre potreby zápisu zrkadlovo otočíme, ale nebudeme ho nijak ďalej oddeľovať ani ukončovať. Aby sme si ukázali, ako bude takýto reťazec výstupu vyzerať dokopy, vezmime si z nášho stroja M časť definície z jeho prechodovej funkcie napríklad majme nasledujúci predpis $\delta(q_2, 0\square\square) = (\square 0, \text{RSS}, q_1)$. Potom reťazec výstupu pre daný vstup bude vyzerať nasledovne □0|□00|10. Pre pomoc s rozlíšením jednotlivých údajov sme ich názorne oddelili zvislou čiarou, ale tieto čiary nie sú súčasťou výstupu a viac ich nebudeme používať. Takýto výstup teda hovorí, že na predposlednú a poslednú pásku sa zapíše v poradí prázdny symbol a symbol 0, prvá páska sa posunie doprava, pričom druhá a tretia zostanú na mieste a stav sa zmení na stav q_1, ktorý je v množine stavov druhý v poradí.

Pridaním tohto reťazca k zápisu vstupu, ako sme si ho skorej definovali, dostávame reťazec zápisu prechodovej funkcie Turingovho stroja a takýto reťazec budeme chcieť používať ako jeho reprezentáciu m. Rozšírením príkladu nášho stroja M aj o výstupy prechodovej funkcie dostávame nasledujúci reťazec

@00/0□□□□00001/1□□□□00001@01@10/0□□□0□0010@.

Tento reťazec hovorí o tom, že štartovací stav má dve možné trojice symbolov načítaných z pások a výstupom z prechodovej funkcie v oboch prípadoch je opätovné zapísanie prázdnych symbolov na druhú a tretiu pásku, zotrvanie prvej, druhej a aj tretej hlavy namieste a zmena stavu v registri na tretí v poradí, označený binárnym číslom 10. Pre druhý stav nemáme definované žiadne trojice, teda ak sa do tohto stavu dostane stroj M tak určite ukončí svoj chod. Pre posledný stav máme definovanú len jednu vstupnú trojicu, kde je jej výstup tvorený zápisom prázdneho symbolu na pracovnú pásku a symbolu 0 na výstupnú pásku, prvá hlava sa posunie doprava a zvyšné hlavy zostanú na mieste a stav sa zmení na druhý v poradí označený binárnym číslom 01. Vidíme tiež, že do tretieho stavu sa stroj môže dostať aj s inou vstupnou trojicou a teda v tomto prípade by stroj ukončil svoj chod s prázdnym reťazcom ako výstupom.

Radi by sme teraz ukázali, že takýto reťazec je v skutočnosti aj reprezentáciou Turingovho stroja. Sformulujme si to do nasledujúceho tvrdenia, ktoré si následne dokážeme.

Tvrdenie 1 (Reprezentácia Turingovho stroja):

Nech M je Turingov stroj (3, Γ = {☐, 0, 1}, ☐, V = {0, 1}, Q, q_0, δ). Potom reťazec m nad abecedou A = {☐, 0, 1, /, @}, opisujúci prechodovú funkciu δ, v tvare vyššie opísanom, je reprezentáciou Turingovho stroja M.

Dôkaz: Prechodovú funkciu každého stroja M si dokážeme zapísať vo forme tabuľky, kde záhlavie stĺpcov tvoria jednotlivé stavy množiny Q a záhlavia riadkov zase všetky možné trojice znakov nad abecedou Γ. Z opisu vieme, že vstupy máme nejako zoradené a je ich konečný počet, zostáva si uvedomiť, že aj všetky možné trojice načítaných symbolov si môžeme usporiadať pomocou nejakého lexikografického zobrazenia nad Γ, samotné zobrazenie pre nás nie je dôležité, podstatné je, že existuje a vieme si nejaké zvoliť. Do buniek v tabuľke budeme zapisovať jednotlivé výstupy, v zmysle zápisu aký sme použili aj pri definovaní reťazca, podľa toho v stĺpci akého stavu sa nachádzame a podľa riadku zase určíme aké symboly sme prečítali z troch pások stroja M. Pokiaľ daná dvojica nemá definovaný žiaden výstup, bunku necháme prázdnu a teda vyplníme len tie bunky, pre ktoré sme výstupný reťazec definovali pri opise tvaru tohto zápisu. Takže vieme, z toho ako je definovaná prechodová funkcia, že takáto tabuľka bude existovať a vzhľadom na dané usporiadania bude táto tabuľka dokonca jednoznačná.

Z toho ako je tabuľka definovaná hneď vidíme, že z nej vieme vyčítať všetky prvky Q a pokiaľ budeme vykonávať prechodovú funkciu tak, že pre každú dvojicu záhlavia stĺpca a riadka definujeme jej výstup podľa obsahu bunky kde sa daný riadok a stĺpec prelínajú alebo danú dvojicu nedefinujeme do prechodovej funkcie ak je bunka prelínania prázdna, vieme presne zrekonštruovať chod stroja M, ktorý daná tabuľka reprezentuje. Dokážeme teda, že danej tabuľke neodpovedá iný stroj M. Už sme si povedali, že každý stroj má až na dané usporiadania množiny Q a množiny trojíc vstupov nad Γ jedinú takúto tabuľku, potrebujeme si však dokázať, že každej tabuľke odpovedá len jeden stroj M. Uvedomme si ako funguje daná tabuľka. Vieme, že pomocou nej vieme zrekonštruovať každý krok prechodovej funkcie a teda pokiaľ dvom rôznym strojom odpovedá rovnaká tabuľka majú aj rovnakú prechodovú funkciu. Rovnako vieme, že množina stavov bude v takýchto strojoch rovnaká. Z rovnakej prechodovej funkcie a množiny stavov plynie, že musia mať rovnaký počet pások, rovnaké abecedy, rovnaké výstupné abecedy a aj rovnaké počiatočné stavy. Teda oba tieto stroje majú rovnakú definíciu a teda sa rovnajú.

Zostáva dokázať, že takúto tabuľku vieme zapísať v tvare reťazca, ako sme si ho opísali vyššie. To nahliadneme tak, že tabuľka kde by sme do záhlavia stĺpcov pred každý stav pridali symbol @, do záhlaví riadkov by sme pred každú trojicu pridali symbol / a pridali by sme na koniec jeden prázdny stĺpec, ktorý by mal v záhlavný symbol @, tak potom je takáto tabuľka ekvivalentná tabuľke, ktorú sme opísali vyššie. Ak si teda jednotlivé stĺpce napíšeme do riadku tak, že začneme záhlavím a budeme postupne pre každú nenulovú bunku v stĺpci zapisovať záhlavie riadku spolu s obsahom tejto bunky a vynechávať budeme tie prázdne dostaneme nami opísaný reťazec. Podobne môžeme postupovať opačne a z reťazca vytvoriť opísanú tabuľku, teda reťazec opisu prechodovej funkcie je reprezentáciou Turingovho stroja *M*. ■

Tento reťazec opisu prechodovej funkcie budeme teda nazývať *reprezentáciou Turingovho stroja* a budeme ho využívať na predávanie stroja *M* ako jeden zo vstupných argumentov do univerzálneho stroja, ktorý si v ďalšej sekcií zavedieme.

2.2 Zavedenie univerzálneho Turingovho stroja

Z názvu sekcie sa dá usúdiť, že v tejto sekcií budeme chcieť popísať a formálne nadefinovať univerzálny Turingov stroj *U*. Nato aby sme konkrétne popísali Turingov stroj je potrebné opísať množinu jeho stavov, spolu s popisom ako bude v týchto stavoch fungovať prechodová funkcia nášho stroja *U*. Na záver sekcie sa pokúsime tento formálny popis zhrnúť do tvrdenia, ktorého dôkaz bude náplňou ďalšej sekcie.

Na začiatok teda uvažujme nasledujúci Turingov stroj

$$U = (5, \Gamma_u = \{\square_u, \square, 0, 1, \triangleright, /, @\}, \square_u, V_u = \{\square, 0, 1, /, @\}, Q_u, q_{0u}, \delta_u).$$

Pričom vstupom do Turingovho stroja *U* bude dvojica (x, m), kde $m \in V_u$ reprezentuje ľubovoľný Turingov stroj $M = (3, \Gamma = \{\square, 0, 1\}, \square, V = \{0, 1\}, Q, q_0, \delta)$, v zmysle ako hovorí Tvrdenie 1 a $x \in V^*$ je vstupný reťazec znakov nad vstupno-výstupnou abecedou stroja *M*, ktorý je zároveň vstupom aj do stroja *U*. Abeceda stroja U obsahuje symbol ▷, ktorý budeme používať na označenie ľavého okraja niektorých pások stroja, jedná sa len o pracovný symbol a nebude súčasťou vstupu ani výstupu stroja U, pomenujme si ho ako štartovací symbol. Okrem toho vidíme, že vstup do stroja U, je tvorený dvojicou reťazcov. Potrebujeme si teda dodefinovať, ako takýto vstup predáme Turingovmu stroju. Pre tieto potreby nám Definícia 1 na str. 4 definuje stroj U a zároveň dodáme, že sa jedná o Turingov stroj s dvoma vstupnými páskami. To znamená, že $k \geq 3$ a prechodová funkcia má tvar δ_u: $Q_u \times \Gamma_u^k \rightarrow Q_u \times \Gamma_u^{k-2} \times \{L, S, R\}^k$. Nech teda náš stroj U, túto definíciu spĺňa a preto môžeme dva vstupy definovať tak, že prvý z dvojice sa zapíše rovnako ako keby bol len jeden na prvú pásku a podobne druhý na druhý pásku, pričom o oboch budeme uvažovať ako o vstupných páskach. Táto úvaha je braná bez ujmy na všeobecnosti, ako hovorí Predpoklad 2. Ďalej v definícií tohto stroja zavedieme použitie štartovacieho symbolu ▷, ktorý budeme mať v štartovacej konfigurácií zapísaný v prvom bloku prechodovej a stavovej pásky, pričom hneď na druhom bloku prechodovej pásky bude začínať druhý vstup stroja U, ostatné časti štartovacej konfigurácie zastávajú nezmenené.

Vidíme, že v našej definícií má stroj U päť pások, ktoré bude využívať. Vo všeobecnosti by sa dalo pracovať s menším počtom pások, existujú možnosti, ako navrhnúť jedno-páskový univerzálny Turingov stroj, ale ako už bolo spomínané, v tejto práci sa chceme zamerať na pochopenie pojmu a toho aby bolo jednoduché vyčítať z popisu stroja ako funguje, takže v našom prípade sa zameriame na pevný počet pások rovný piatim. Predpoklad 2 hovorí, že stroj M, ktorý budeme chcieť simulovať, môžeme uvažovať ako troj-páskový, kde jeho prvá páska sa nazýva vstupná, druhá zase pracovná a tretia výstupná páska. Tieto pomenovania si ponecháme aj v našom univerzálnom stroji U, kde takto nazveme v poradí prvú, štvrtú a poslednú teda piatu pásku. Tieto pásky budú teda spĺňať presne rovnaký účel ako v stroji M, čiže na vstupnej sa bude nachádzať reťazec x a nebudeme na ňu môcť zapisovať, na pracovnú si budeme zapisovať medzivýsledky, presne tak ako by to robil stroj M a na výstupnú zapíšeme výstup tak, ako hovorí Definícia 2 na str. 6 a tak ako by to spravil aj stroj M.

Zostávajúce dve pásky využijeme na chod stroja U a vykonávanie jeho prechodovej funkcie, teda simuláciu stroja M. Prvú z týchto pások, druhú vstupnú pásku stroja U, si označíme ako prechodovú pásku, pretože na tejto páske sa bude nachádzať druhý z dvojice vstupov stroja U, teda reťazec m reprezentujúci stroj M. Na túto pásku nebude stroj U môcť nič zapisovať, keďže sa jedná o jednu zo vstupných pások stroja U. Posledná nepoužitá páska sa bude pracovne označovať ako stavová páska a to v tom zmysle, že sa na nej bude zapísaná binárna reprezentácia stavu, ktorý by bol inak zapísaný v premennej aktuálneho stavu q stroja M. Okrem iného budeme stavovú pásku využívať aj ako pracovnú pásku stroja U, na ktorú budeme zapisovať pomocné hodnoty pri vykonávaní prechodovej funkcie δ_u, spojené so simuláciou prechodovej funkcie δ stroja M. Obe tieto pásky budú obsahovať symboly z abecedy Turingovho stroja U zatiaľ, čo o troch pôvodných môžeme povedať, že budú obsahovať len znaky abecedy stroja M.

Stroj U bude pracovať s abecedou označenou Γ_u, ktorá ako vidíme, obsahuje celú abecedu Γ, využívajúcu Predpoklad 1. Abeceda Γ_u okrem toho obsahuje štyri symboly naviac od abecedy Γ a dva z týchto symbolov (/, @) sú potrebné z hľadiska použitia reťazca na reprezentáciu stroja. Ďalší z týchto symbolov používame na rozlíšenie prázdneho symbolu stroja M od prázdneho symbolu stroja U, čiže prázdny symbol stroja U označíme symbolom $\square_u$ a prázdny symbol stroja M zase symbolom $\square$. Posledný nový symbol $\triangleright$ budeme používať na označenie začiatku dvoch pások, konkrétne prechodovej a stavovej pásky.

Ďalej by sme radi popísali jednotlivé stavy a celú konečnú množinu stavov stroja U. Najskôr si tieto stavy pomenujme názvami a zamyslíme sa nad tým ako tieto stavy budú fungovať. Označme teda Q_u = {START, COPY, FINDSTATE, READ, WRITE, MOVE, CHANGESTATE}. Stavy v Q_u sme si zatiaľ označili len anglickými slovami. Učinili sme tak preto, aby sme získali lepší intuitívny náhľad, ako budú tieto stavy pracovať a čo budú v chode stroja reprezentovať, bez toho aby sme formálne definovali prechodovú funkciu. Potom tieto názvy môžeme previesť do našej binárnej reprezentácie alebo môžeme uvažovať, že dokážeme ukladať aktuálny stav práve pomocou týchto slov a môžeme ich ďalej v práci pomenovávať týmito anglickými názvami, ktoré majú lepší intuitívny charakter.

Prvým stavom v zozname je stav START, čo používame ako označenie štartovacieho stavu, takže má charakter podobný ako v samotnej definícií

Turingovho stroja. Jediný rozdiel je, že za vstupnú pásku považujeme aj druhú teda prechodovú pásku, na ktorej sa už pred samotnou činnosťou stroja nachádza reprezentácia stroja M v podobe určeného reťazca m. Takže tento stav bude slúžiť len na označenie začiatku fungovania stroja.

Ďalšie stavy sú už konkrétne pre náš univerzálny Turingov stroj U, a teda ich intuitívny náhľad sa pokúsime podať podrobnejšie. Prvý z týchto stavov sa nazýva COPY a teda presne toto bude tento stav vykonávať. Bude kopírovať určitý obsah prechodovej pásky na stavovú, konkrétne skopíruje reprezentáciu prvého teda štartovacieho stavu stroja M na stavovú pásku, aby ho pri simulácií pokladal stroj U za aktuálny a vykonal ho ako prvý. Po ukončení kopírovania budeme pokračovať hľadaním aktuálneho stavu na prechodovej páske.

Ďalší z týchto stavov sme si označili ako FINDSTATE, čiže v preklade „Nájdi stav". Tým chceme naznačiť, že tento stav bude slúžiť k hľadaniu aktuálneho stavu zapísaného na stavovej páske medzi stavmi, ktoré slúžia ako vstup do prechodovej funkcie stroja M, zapísanými na prechodovej páske. Tieto stavy sú zapísané v našej bitovej reprezentácií v konkrétnej dĺžke, ktorá je však nepodstatná pretože na oddelenie stavu od ďalších zápisov na tejto páske použijeme symbol / prípadne symbol @ a teda konkrétnu dĺžku, ktorá môže byť pre každý stroj iná, nebudeme vôbec využívať. Takže hľadanie budeme vykonávať tak, že vyhľadáme symbol @ na prechodovej páske. Potom porovnáme symboly bezprostredne za týmto symbolom, so stavom uloženým na stavovej páske, až pokým nenarazíme na symbol /. V tomto momente sme ho našli. Ale ak nájdeme rozdiel medzi symbolmi, vraciame sa k hľadaniu ďalšieho symbolu @. Keďže takto musia byť v reprezentácií zapísané všetky stavy, do ktorých sa stroj M môže dostať, tak vieme, že tento stav musíme nájsť. Akonáhle nájdeme aktuálny stav, tak pokračuje chod stroja v stave READ. Jediný rozdiel nastáva, pokiaľ sme prešli na koniec stavu a narazili sme na stav, ktorý je ukončený symbolom @. V takomto prípade nedefinujeme výstup prechodovej funkcie stroja U a ukončíme jeho chod s tým, že výstup stroja M, ktorý sme simulovali, sa nachádza podľa definície na poslednej páske stroja U.

Preklad názvu stavu READ napovedá, že v priebehu tohto stavu sa budeme snažiť simulovať čítanie z pások stroja M. Toto čítanie bude mať nasledovnú podobu. V reťazci si nájdeme najbližší znak / vpravo od našej aktuálnej pozície a za ním budeme postupne porovnávať trojicu symbolov s tým čo máme zapísané na aktuálnych pozíciách hláv. V poradí prvý symbol porovnáme so zápisom na vstupnej páske, druhý symbol so symbolom načítaným z pracovnej pásky a posledný porovnáme s výstupnou páskou. Ak sú všetky tri hodnoty rovnaké, tak sme pre aktuálny stav našli úplnú dvojicu vstupov do prechodovej funkcie a môžeme pokračovať jej aplikovaním pomocou stavov WRITE, MOVE a CHANGESTATE, v tomto poradí. Ak narazíme na nezhodné symboly, pokračujeme v hľadaní ďalšieho symbolu / v poradí pre daný aktuálny stav. Keď z prechodovej pásky načítame symbol @, tak pre daný stav nemáme nadefinovanú prechodovú funkciu stroja M s danou trojicou symbolov. Takže nedefinujeme pre takýto prípad prechodovú funkciu stroja U, čím ukončíme v takomto prípade chod stroja U, s výstupom na poslednej páske.

Samotné aplikovanie prechodovej funkcie budú zabezpečovať už spomenuté tri stavy. Tieto tri stavy budú pracovať nad reťazcom symbolov, umiestnenom hneď za trojicou načítaných symbolov, ktoré sme porovnávali v stave READ. Prvý z týchto stavov, do ktorého sa dostaneme sa nazýva WRITE a preklad znamená „Zapíš“, pričom presne toto bude tento stav vykonávať. Prvé dva symboly z reťazca výstupu prechodovej funkcie tvoria nové symboly, ktoré majú byť zapísané na posledné dve pásky stroja M a presne toto teda stroj U v tomto stave vykoná. Okrem prechodovej, pracovnej a výstupnej pásky bude využívať aj stavovú pásku, na ktorú si mimo zapísaný stav bude značiť na akú pásku aktuálne zapisoval. Využije nato symboly nad abecedou Γ, konkrétne symboly si rozoberieme pri formálnej definícií prechodovej funkcie. Dôležité však je, že na základe symbolu prečítaného zo stavovej pásku bude vedieť kam má načítaný symbol z prechodovej pásky zapísať, či na pracovnú alebo na výstupnú pásku a okrem toho zapíše aj nový znak na stavovú pásku pre indikáciu voľby v ďalšom kroku. Pomocou tohto symbolu na stavovej páske okrem toho budeme rozhodovať, či sme už so zápisom skončili alebo nie.

Keď vyhodnotíme koniec zápisu prepneme sa do stavu MOVE. V tomto stave sa bude jednať o pohyb hláv, presne podľa predpisu prechodovej funkcie stroja M. V reťazci výstupu sme si nadefinovali tri špeciálne znaky, ktoré určujú, ktorým smerom konkrétnu hlavu posunieme, prípadne ponecháme na mieste (pre pripomenutie: {1 - doľava, 0 - nezmeniť pozíciu, □ - doprava}). V reťazci sa teda ďalej nachádza trojica symbolov, ktorá definuje, kam sa majú v poradí posunúť vstupná, pracovná a výstupná hlava. Podobne ako v stave WRITE aj tu budeme využívať pomoc stavovej pásky pre značenie, ktorú hlavu máme ďalej posúvať a či sme už skončili s posunom. Teda načítame vždy symbol z prechodovej a stavovej pásky a na ich základe sa rozhodneme, ktorú hlavu a kam posunieme, prípadne, či sme už ukončili posun a prejdeme na stav CHANGESTATE.

Tento stav, CHANGESTATE, bude pracovať s poslednými pamäťovými blokmi v reťazci výstupu a koniec jeho vykonávania budeme rozlišovať symbolom @ alebo /, ktoré označujú koniec jedného reťazca výstupu a teda jasne vieme, kedy sme so zmenou aktuálneho stavu skončili. Zápis nového stavu bude teda spočíva, v prepise načítaného symbolu z prechodovej pásky na požadované miesto na stavovej páske, pričom tvar zápisu nového stavu v reťazci výstupu má formu otočeného binárneho zápisu stavov, tak ako sme si ho skorej definovali. Takže jeho prepis do stavovej pásky je jednoduchý. Budeme postupovať v opačnom smere, teda zapisovať ho od konca. Pričom môžeme rozhodovať aj na základe symbolu načítanom zo stavovej pásky, kedy pokiaľ načítame štartovací symbol ▷ už sme určite ukončili zápis stavu a môžeme sa prepnúť späť do stavu FINDSTATE. V tomto stave by sa chod stroja ukončil, ak sme práve zapísali bitovú reprezentáciu stavu, pre ktorý nemáme definovanú prechodovú funkciu.

Takto by sme radi nadefinovali množinu stavov tak, aby fungovali podľa intuitívneho náhľadu a takýmto spôsobom simulovali chod stroja M. Avšak pre niektoré tieto stavy bude ešte pred ich začatím potrebné posúvať jednotlivé hlavy na vhodné pozície. Preto si za týmto účelom dodefinujeme ďalšie stavy, ktoré budeme nazývať pomocné. Takže reálna množina stavov stroja U bude vyzerať nasledovne Q_u = {START, COPY, FINDSTATE, READ, WRITE, MOVE, CHANGESTATE, LEFT, LOCATE@, STATELEFT, LOCATE/}. Na tieto pomocné stavy nebudeme poskytovať intuitívny náhľad, pretože pôjde len o technické stavy, ktoré budú zabezpečovať správny chod staja U. Ich konkrétny popis teda bude opísaný vo formálnej definícií prechodovej funkcie δ_u, ktorú položíme na nasledujúcich stranách tejto sekcie.

Prechodovú funkciu Turingovho stroja U budeme teda definovať nasledujúcim výpisom všetkých možných situácií, v ktorých sa môže stroj U ocitnúť, spoločne s výpisom tabuliek výstupov prechodovej funkcie stroja U, pre jednotlivé stavy. Budeme uvažovať q ako označenie aktuálneho stavu stroja U a y nám bude reprezentovať reťazec načítanej pätice symbolov z piatich pások stroja U. V tabuľke budeme vždy uvažovať len také vstupy, pre ktoré je prechodová funkcia definovaná, až na prípady, keď sa do konkrétnych vstupov môže stroj U dostať a chceme explicitne uviesť, že v takomto prípade rátame s ukončením chodu stroja, pokiaľ takáto situácia nastane. Okrem toho v tejto tabuľke budeme využívať takzvané neurčité symboly, ktoré budú zastupovať vybrané symboly abecedy Γ_u. Položme $N = \{\cancel{\sqcup_u}, \cancel{\sqcup}, \cancel{0}, \cancel{1}, \cancel{\rhd}, \cancel{/}, \cancel{@}, ?, s, x, \cancel{x}, (,), \&\}$. Potom budeme uvažovať, že $y \in (\Gamma_u \cup N)^*$ je vstupom do δ_u. Pričom ak $a \in \Gamma_u \cup \{x\}$ potom symbol $\cancel{a} \in N$, hovorí, že z danej pásky môžeme načítať ľubovoľný symbol okrem symbolu a. Symbolom ? budeme reprezentovať, že z danej pásky môžeme načítať ľubovoľný symbol nad Γ_u. Symbolom s zase povieme, že na danú pásku zapíšeme ten istý symbol, ktorý sme z nej načítali. Symbol x bude premenná, ktorá bude označovať symbol načítaný z danej pásky. Symboly *(*, *)* a *&* budú používané spoločne a spájať viaceré podmienky na symbol z danej pásky. Ako je vidieť, niektoré symboly patriace N, použijeme aj pri výpise výstupu prechodovej funkcie. Definujeme teda δ_u nasledovným spôsobom.

Ak je aktuálny stav q rovný stavu START, tak potom ak je symbol načítaný z prechodovej a stavovej pásky rovný znaku ▷ tak posuň ich hlavy doprava, inak s inými hlavami nehýb, nič nezapisuj a zostaň v stave START. Ak je z prechodovej pásky načítaný symbol @ tak posuň túto hlavu doprava, nič nezapisuj, iné hlavy neposúvaj a zmeň stav na COPY. Tabuľka pre stav START vyzerá nasledovne.

y	$\delta_u(\mathbf{START}, y)$
?▷▷??	(sss, SRRSS, START)
?@???	(sss, SRSSS, COPY)

Tabuľka 1: Prechodová funkcia stavu START

Ak je stroj U v stave COPY, tak ak z prechodovej pásky načítame symbol 0 alebo 1, tak ho zapíš na stavovú pásku a obe hlavy posuň doprava, na iné pásky nič nezapisuj, ani neposúvaj ich hlavy a zostaň v stave COPY. Ak načítame z prechodovej pásky symbol / alebo @, tak zapíš na stavovú pásku symbol /, nič iné nezapisuj, posuň prechodovú a stavovú hlavu doľava a zmeň stav na pomocný stav LEFT. Popísané hodnoty zapíšeme do nasledujúcej tabuľky.

$\boldsymbol{y}$	$\boldsymbol{\delta_u}$**(COPY,** $\boldsymbol{y}$**)**
?0???	(0ss, SRRSS, COPY)
?1???	(1ss, SRRSS, COPY)
?/???	(/ss, SLLSS, LEFT)
?@???	(/ss, SLLSS, LEFT)

Tabuľka 2: Prechodová funkcia stavu COPY

Ak je stav stroja *U* rovný stavu LEFT, tak načítaj symbol z prechodovej a stavovej pásky. Ak sú oba rovné štartovaciemu symbolu, tak ich obe posuň doprava, nič nezapisuj a zmeň aktuálny stav na pomocný stav LOCATE@. Inak nič nezapisuj a ani nemeň stav, ale posuň obe hlavy doľava. Presný popis udáva nasledujúca tabuľka.

$\boldsymbol{y}$	$\boldsymbol{\delta_u}$**(LEFT,** $\boldsymbol{y}$**)**
?$\triangleright\triangleright$??	(sss, SRRSS, LOCATE@)
?$\cancel{\triangleright}\cancel{\triangleright}$??	(sss, SLLSS, LEFT)
?$\cancel{\triangleright}\triangleright$??	(sss, SLSSS, LEFT)
?$\triangleright\cancel{\triangleright}$??	(sss, SSLSS, LEFT)

Tabuľka 3: Prechodová funkcia stavu LEFT

Ak je v registri stroja *U* uložený stav LOCATE@, tak potom načítaj symbol z prechodovej pásky. Ak je tento symbol rovný @, tak posuň túto hlavu doprava, iné hlavy neposúvaj, nič nezapisuj a zmeň stav na FINDSTATE. Ak je načítaný symbol iný, tak len posuň prechodovú hlavu doprava a nič iné nemeň. Nasledujúca tabuľka udáva správanie prechodovej funkcie v tomto stave.

$\boldsymbol{y}$	$\boldsymbol{\delta_u}$**(LOCATE@,** $\boldsymbol{y}$**)**
?@???	(sss, SRSSS, FINDSTATE)
?$\cancel{@}$???	(sss, SRSSS, LOCATE@)

Tabuľka 4: Prechodová funkcia stavu LOCATE@

V stave FINDSTATE prečítaj symboly z prechodovej a stavovej pásky. Ak je symbol prečítaný z prechodovej aj stavovej pásky rovný /, tak posuň stavovú hlavu doprava, iné hlavy neposúvaj, nič nezapisuj a zapíš nový stav LOCATE/. Ak je na prechodovej páske zapísaný symbol @ a na stavovej páske symbol /, tak v tomto prípade nedefinujeme výstup prechodovej funkcie, ale dôjde k ukončeniu chodu stroja, s výstupom nachádzajúcim sa na poslednej páske, tak ako sme ho definovali. Ak sú oba načítané symboly rovné 0 alebo 1, tak obe pásky posuň doprava, ale nič nezapisuj a stav nemeň. Ak sú načítané symboly rôzne a symbol z prechodovej pásky rôzny od @, tak posuň prechodovú hlavu doprava a stavovú hlavu doľava, nič nezapisuj a zmeň stav na pomocný stav STATELEFT. Zapíšme tieto pravidlá do nasledujúcej tabuľky, reprezentujúcej prechodovú funkciu stavu FINDSTATE stroja *U*.

y	δ_u**(FINDSTATE,** y**)**
?//??	(sss, SSRSS, LOCATE/)
?@/??	Ukončíme chod stroja *U*
?00??	(sss, SRRSS, FINDSTATE)
?11??	(sss, SRRSS, FINDSTATE)
?01??	(sss, SRLSS, STATELEFT)
?10??	(sss, SRLSS, STATELEFT)

Tabuľka 5: Prechodová funkcia stavu FINDSTATE

Pre pomocný stav STATELEFT definujeme prechodovú funkciu tak, že budeme načítavať symbol zo stavovej pásky a ak je rovný štartovaciemu symbolu $\triangleright$, tak posunieme stavovú hlavu doprava, nič nezapisujeme, s inými hlavami nehýbeme a zmeníme stav na ďalší pomocný stav LOCATE@. Ak ale symbol načítaný zo stavovej pásky, nie je rovný tomuto symbolu, tak posunieme stavovú hlavu doľava a nič iné nemeníme. Definujme teda pre tento stav prechodovú funkciu pomocou nasledujúcej tabuľky.

y	δ_u**(STATELEFT,** y**)**
??$\triangleright$??	(sss, SSRSS, LOCATE@)
??$\not\triangleright$??	(sss, SSLSS, STATELEFT)

Tabuľka 6: Prechodová funkcia stavu STATELEFT

Pre stav LOCATE/ budeme načítavať symbol z prechodovej pásky a budeme sa pýtať, či je tento symbol rovný symbolu /. Ak áno, tak posunieme prechodovú hlavu doprava, na stavovú pásku zapíšeme prázdny symbol $\square_u$, inak nič nezapisujeme, s inou hlavou nehýbeme a do registra stavov zapíšeme stav READ. Ak však nie je rovný tomuto symbolu a ani symbolu @, tak iba posunieme prechodovú hlavu doprava, nič nezapisujeme a stav nemeníme. Ak ale narazíme na symbol @, tak vieme, že v danom stave a trojici nemáme definovanú prechodovú funkciu, takže nedefinujeme výstup a ukončíme chod stroja. Popis sa nachádza v nasledujúcej tabuľke.

$\boldsymbol{y}$	$\boldsymbol{\delta_u(\mathrm{LOCATE/}, y)}$
?/???	(□$_u$ss, SRSSS, READ)
?@???	Ukončíme chod stroja U
?(~~/~~ & ~~@~~)???	(sss, SRSSS, LOCATE/)

Tabuľka 7: Prechodová funkcia stavu LOCATE/

Ak sa stroj nachádza v stave READ, tak nás bude zaujímať symbol načítaný zo stavovej pásky. Ak je na stavovej páske uložený prázdny symbol □$_u$, tak načítame symbol zo vstupnej pásky a porovnáme ho zo symbolom na prechodovej páske, ak sa rovnajú zapíšeme na stavovú pásku symbol 0, posunieme prechodovú pásku doprava, zostaneme v stave READ a nič iné nemeníme. Ak je na stavovej páske zapísaný symbol 0, tak porovnáme symboly z pracovnej a prechodovej pásky a ak sa rovnajú, tak posunieme prechodovú hlavu doprava, zapíšeme na stavovú pásku symbol 1, zostaneme v stave READ a nič iné nemeníme. Ak zo stavovej pásky načítame symbol 1, tak porovnávame symboly z prechodovej a výstupnej pásky a pokiaľ sa rovnajú, tak zapíšeme na stavovú pásku symbol 0, posunieme prechodovú hlavu doprava a zmeníme stav na WRITE, zatiaľ čo iné pásky neposúvame a ani nič iné nezapisujeme. Pokiaľ niekedy narazíme na nerovnosť, v hociktorom z týchto troch prípadov, tak posunieme prechodovú hlavu doprava a zmeníme stav na LOCATE/, pričom nič iné v aktuálnej konfigurácii stroja U nemeníme. Pre tento stav definujeme nasledujúcu tabuľku.

$\boldsymbol{y}$	$\boldsymbol{\delta_u(\mathrm{READ}, y)}$
xx□$_u$??	(0ss, SRSSS, READ)
?x0x?	(1ss, SRSSS, READ)
?x1?x	(0ss, SRSSS, WRITE)
x~~x~~□$_u$??	(sss, SRSSS, LOCATE/)
?x0~~x~~?	(sss, SRSSS, LOCATE/)
?x1?~~x~~	(sss, SRSSS, LOCATE/)

Tabuľka 8: Prechodová funkcia stavu READ

Pre stav WRITE definujeme prechodovú funkciu podľa nasledujúcich pravidiel. Načítame symbol zo stavovej pásky a ak je rovný symbolu 0, tak zapíšeme symbol z prechodovej pásky na pracovnú pásku, na stavovú pásku zapíšeme symbol 1 a prechodovú pásku posunieme doprava, nič iné nemeníme. Ak je na stavovej páske uložený symbol 1, tak zapíšeme na výstupnú pásku symbol z prechodovej pásky, na stavovú pásku zapíšeme prázdny symbol $\square_u$, prechodovú hlavu posunieme doprava, nič iné nezapisujeme, iné hlavy neposúvame a aktuálny stav zmeníme na MOVE. Nasleduje tabuľka tohto stavu.

$\boldsymbol{y}$	$\boldsymbol{\delta_u(\mathrm{WRITE}, y)}$
?x0??	(1xs, SRSSS, WRITE)
?x1??	($\square_u$sx, SRSSS, MOVE)

Tabuľka 9: Prechodová funkcia stavu WRITE

Stav MOVE bude opäť načítavať symbol zo stavovej a prechodovej pásky. Ak je táto dvojica rovná v poradí prázdnemu symbolu $\square_u$ stroja U a prázdnemu symbolu $\square$ stroja M, tak na základe toho ako sme si nadefinovali množinu reprezentujúcu jednotlivé posuny, posunieme vstupnú hlavu doprava, prechodovú hlavu doprava, aby sme mohli v ďalšom kroku pokračovať s posunom ďalšej hlavy, na stavovú pásku zapíšeme symbol 0, rozhodujúci aká páska sa bude posúvať v ďalšom kroku a inak nič nemeníme. Ak je táto dvojica zase rovná prázdnemu symbolu $\square_u$ a 1 v poradí, tak posúvame vstupnú hlavu doľava, prechodovú hlavu vpravo, na stavovú pásku zapíšeme 0 a inak zase nič nemeníme. Ak je táto dvojica rovná prázdnemu symbolu $\square_u$ a 0 tak posúvame len prechodovú hlavu doprava, na stavovú pásku zapíšeme 0 a inak nič nezmeníme. Podobne postupujeme pre dvojice, kde je znakom zo stavovej pásky 0, s tým rozdielom, že na stavovú pásku v tomto prípade zapisujeme symbol 1 namiesto symbolu 0 a posúvame pracovnú hlavu namiesto vstupnej. Ak je prvým z dvojice zase symbol 1, tak postupujeme zase podobne ako v opísaných prípadoch pre jednotlivé dvojice s rozdielom, že posúvame výstupnú hlavu namiesto vstupnej alebo pracovnej, na stavovú pásku zapíšeme prázdny symbol $\square_u$ namiesto symbolov 0 alebo 1 a okrem toho zmeníme v tomto prípade aj to, že stavovú pásku posunieme doľava a tiež zmeníme aktuálny stav stroja U na CHANGESTATE. Pre stav MOVE prechodovej funkcie stroja U teda definujeme tabuľku opisujúcu predpis vstupov a prislúchajúcich výstupov pre prechodovú funkciu nasledujúcim spôsobom.

y	δ_u(MOVE, y)
?□$□_u$??	(0ss, RRSSS, MOVE)
?1$□_u$??	(0ss, LRSSS, MOVE)
?0$□_u$??	(0ss, SRSSS, MOVE)
?□0??	(1ss, SRSRS, MOVE)
?10??	(1ss, SRSLS, MOVE)
?00??	(1ss, SRSSS, MOVE)
?□1??	($□_u$ss, SRLSR, CHANGESTATE)
?11??	($□_u$ss, SRLSL, CHANGESTATE)
?01??	($□_u$ss, SRLSS, CHANGESTATE)

Tabuľka 10: Prechodová funkcia stavu MOVE

V stave CHANGESTATE sa budeme pýtať nato, aký symbol sme načítali zo stavovej pásky. Ak je tento symbol rôzny od symbolu / a na prechodovej páske sme zároveň načítali iný symbol ako @ alebo /, tak zapíšeme na stavovú pásku symbol z prechodovej pásky, posunieme prechodovú pásku doprava, stavovú pásku doľava a nič iné nemeníme. Ak symbol na prechodovej páske je rovný symbolu / alebo symbolu @ a na stavovej páske je štartovací symbol, tak posunieme prechodovú hlavu doľava, nič nezapisujeme, žiadnu inú hlavu neposúvame a zmeníme stav na LEFT. Ale ak symbol na stavovej páske je rovný symbolu /, tak nič nezapisujeme, posunieme stavovú hlavu doľava, iné hlavy neposúvame a zostaneme v stave CHANGESTATE. Zapíšeme tento predpis pre stav CHANGESTATE do nasledujúcej tabuľky.

y	δ_u(CHANGESTATE, y)
??/??	(sss, SSLSS, CHANGESTATE)
?/▷??	(sss, SLSSS, LEFT)
?@▷??	(sss, SLSSS, LEFT)
?1$\not/$??	(1ss, SRLSS, CHANGESTATE)
?0$\not/$??	(0ss, SRLSS, CHANGESTATE)

Tabuľka 11: Prechodová funkcia stavu CHANGESTATE

Takto opísaný Turingov stroj U by sme radi označili za univerzálny. Nato však potrebujeme túto univerzálnosť dokázať. Sformulujme si teda toto tvrdenie, pričom jeho dôkaz bude náplňou sekcie 2.3.

Tvrdenie 2 (Univerzálny Turingov stroj):

Nech $U = (5, \Gamma_u = \{\square_u, \square, 0, 1, \triangleright, /, @\}, \square_u, V_u = \{\square, 0, 1, /, @\}, Q_u, q_{0u}, \delta_u)$ *je Turingov stroj, kde* Q_u = {START, COPY, FINDSTATE, READ, WRITE, MOVE, CHANGESTATE, LEFT, LOCATE@, STATELEFT, LOCATE/}, q_{0u} = START, *prechodová funkcia* δ_u *je definovaná v Tabuľkách 1 až 11 a U je Turingov stroj definovaný s dvomi vstupmi. Potom U je univerzálny Turingov stroj.*

2.3 Dôkaz univerzálnosti

Táto sekcia podá náhľad na dôkaz univerzálnosti nášho stroja *U*, ktorý sme si zadefinovali v predchádzajúcej sekcii. Tvrdenie 2 hovorí o výsledkoch z predchádzajúcej sekcie a ako sme povedali, táto sekcia sa bude venovať dôkazu tohto tvrdenia. Samotný dôkaz univerzálnosti bude pozostávať z niekoľkých pomocných viet a tvrdení, ktoré dokopy budú tvoriť požadovaný dôkaz. Začnime teda nasledujúcou definíciou, ktorá podá náhľad na popis ľubovoľného Turingovho stroja v určitom okamihu jeho chodu.

Definícia 5 (Okamžitý popis):

Nech $M = (k, \Gamma, \square, V, Q, q_0, \delta)$ *je Turingov stroj. Povieme, že* $u = a_1a_2...a_ki_1i_2...i_kq$, *kde* $a_j \in \Gamma^*$, *dĺžka* a_j *je rovná* $l_j \in \mathbb{N}$ *a* $i_j \in \mathbb{N}$ *pre* $\forall j \in \{1, 2, \dots, k\}$, *je okamžitý popis stroja M pre vstup* $x \in V^*$ *v nejakom okamihu jeho chodu, pokiaľ aktuálny stav stroja M je rovný* $q \in Q$, *pre* $\forall j \in \{1, 2, \dots, k\}$ *platí, že* i_j *je index pamäťového bloku j-tej pásky, na ktorý ukazuje jej hlava a zároveň* a_j *je reťazec nachádzajúci sa na prvých* l_j *pamäťových blokoch j-tej pásky, pričom pamäťový blok na* l_j*-tej pozícií tej istej pásky neobsahuje prázdny symbol a na pozíciách* $l_j + 1, l_j + 2, ...$ *rovnakej pásky sa nachádzajú prázdne symboly, ak sú na j-tej páske len prázdne symboly definujeme* a_j *ako prázdny reťazec* ($a_j = \square$).

Ako prvé si môžeme všimnúť, že v každom okamihu chodu, ľubovoľného stroja *M*, pre každý možný vstup *x*, bude pre jeho okamžitý popis platiť, že $a_1 = x$. Ďalej nám Predpoklad 1 a Predpoklad 2 hovoria, že bez ujmy na všeobecnosti, si môžeme definovať stroj $M = (3, \Gamma = \{\square, 0, 1\}, \square, V = \{0, 1\}, Q, q_0, \delta)$ a považovať ho za ľubovoľný stroj. Radi by sme teda nahliadli, ako bude vyzerať okamžitý popis stroja *U*, ktorý bude odpovedať ekvivalentnému popisu stroja *M*, ktorý má *U* simulovať. Sformulujme to do nasledujúcej vety a dokážme si ju.

Veta 1 (Okamžitý popis stroja *U*):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Potom je reťazec $u = xa_2a_3i_1i_2i_3q$ *okamžitým popisom stroja M pre vstup x v nejakom okamihu jeho chodu, práve vtedy keď nasledujúci reťazec* $u^U = x\tilde{m}\tilde{q}a_2a_3i_122i_2i_3q_u$*, kde* $\tilde{m} = \triangleright|m$*,* $\tilde{q} = \triangleright|q|/$*, | značí zreťazenie znakov,* $q_u = LOCATE@ \in Q_u$*, je okamžitým popisom stroja U pre vstup (x, m) v nejakom okamihu jeho chodu.*

Dôkaz: Na začiatok uvažujme množinu popisov stroja *U*, v každom okamihu jeho chodu. Nechajme v tejto množine len tie popisy, pre ktoré platí, že aktuálny stav je LOCATE@ a indexy prechodovej a stavovej pásky sú rovné 2. Z toho ako sme definovali prechodovú funkciu stroja *U* vieme, že takáto množina je neprázdna. Môžeme z nej vybrať len také, kde platí, že obsah stavovej pásky je rovný $\tilde{q}$, to vieme z toho, že stroj *M* sa v stave *q* nachádza práve vtedy, keď je zapísaný na stavovej páske stroja *M* a teda aj takáto množina je neprázdna. Podobne môžeme nahliadnuť to, že keď vyberieme z tejto množiny len popisy kde sú indexy vstupnej, pracovnej a výstupnej hlavy ekvivalentné, tak je neprázdna a zredukovaním na rovné obsahy pracovnej a výstupnej pásky dostávame žiadaný popis stroja *U*, ktorý vieme, že existuje, z toho ako je stroj *U* nadefinovaný. Keďže dané podmienky boli ekvivalentné, dokázali sme obe implikácie. ■

Na základe notácie definovanej v prechádzajúcej vete, budeme hovoriť, že popisy u^U stroja *U* a *u* stroja *M* si odpovedajú. Ak predpokladáme, že pre popis *u*, je definovaná prechodová funkcia stroja *M* a po jej vykonaní sa dostane stroj *M* napríklad do stavu *v*, tak by sa nám hodilo, keby pre stroj *U* platilo, že sa dostane do stavu v^U, ktorý odpovedá stavu *v*. To platí ako si ukážeme v nasledujúcej vete.

Veta 2 (Prechod popisov):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Potom nech reťazec $u = xa_2a_3i_1i_2i_3q$ *je okamžitý popis stroja M pre vstup x v nejakom okamihu jeho chodu a reťazec* u^U *je odpovedajúcim popisom stroja U pre vstup (x, m) v nejakom okamihu jeho chodu. Ak platí, že M sa dostane po jednom kroku jeho prechodovej funkcie do popisu* $v = x\hat{a}_2\hat{a}_3\hat{\imath}_1\hat{\imath}_2\hat{\imath}_3\hat{q}$*, potom sa stroj U dostane po konečnom počte krokov jeho prechodovej funkcie do odpovedajúceho popisu* v^U*.*

Táto veta sa nám bude hodiť pri dokazovaní univerzálnosti stroja *U*. Pre jej dôkaz si naformulujeme ďalšie pomocné tvrdenia o chode stroja *U*. Začneme tvrdením o vykonaní stavu START, ktoré síce nie je potrebné pri dokazovaní práve naformulovanej vety, ale bude sa hodiť pri celkovom dôkaze univerzálnosti.

Tvrdenie 3 (Vykonanie stavu START):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis U je $u = x\tilde{m}\triangleright a_2 a_3 11111 q_u$, *kde* q_u = *START, potom sa U dostane v konečnom počte krokov do popisu* $v = x\tilde{m}\tilde{q}a_2a_3 12211 \tilde{q}_u$, *kde* $\tilde{q}_u$ = *LOCATE@ a obsah prechodovej pásky je* $\tilde{q} = \triangleright|q_0|/$, *kde* q_0 *je štartovací stav stroja M.*

Dôkaz: Vieme, že môžeme predpokladať $a_2 = a_3 = \square$, pretože tak je definovaná štartovacia konfigurácia a stroj *U* sa prvýkrát nachádza v stave START práve v tomto momente. Vidíme, že prvý krok stroja *U*, v tomto stave je posunúť prechodovú a stavovú hlavu doprava takže, sa stroj *U*, dostane do popisu $x\tilde{m}\triangleright a_2a_3 12211 q_u$. V tomto momente načíta z prechodovej pásky symbol @, pretože tak je definovaný reťazec *m*, preto zmení index prechodovej pásky na 3 a prepne sa do stavu COPY.

V jednotlivých krokoch stavu COPY bude zaradom kopírovať znaky z prechodovej pásky na stavovú, pričom tieto znaky sa rovnajú bitovej reprezentácií stavu q_0, pretože tak sme si nadefinovali reťazec *m*. Toto kopírovanie skončí, keď z prechodovej pásky načítame symbol / alebo @, v takomto momente zapíše stroj *U* na stavovú pásku symbol / a vidíme, že obsah stavovej pásky je teda rovný $\tilde{q} = \triangleright|q_0|/$, a tiež sa stroj prepne do stavu LEFT.

V nasledujúcich krokoch bude stroj *U*, len posúvať prechodovú a stavovú hlavu doľava až pokiaľ nenarazí na štartovacie symboly, označujúce jej začiatok a následne ich posunie doprava, spolu so zmenou stavu na LOCATE@. Čo je hľadaný popis *v* stroja *U*. ■

Ako ďalšie by sme radi nahliadli, že sa zo stavu LOCATE@ vždy dostaneme do stavu LOCATE/. Tieto stavy sú síce len pomocné, ale ako vidíme, označujú začiatok hľadania jednotlivých prvkov z dvojice vstupov na reťazci *m* a preto má zmysel zaoberať sa prechodom práve medzi týmito dvoma stavmi.

Tvrdenie 4 (Nájdenie aktuálneho stavu):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis U je $u = x\tilde{m}\tilde{q}a_2a_3i_122i_2i_3q_u$, *kde* q_u = *LOCATE@ a* $\tilde{q} = \triangleright|q|/$, *kde q je aktuálny stav stroja M, predpokladáme, že M pre odpovedajúci popis neukončí svoj chod, potom sa U dostane v konečnom počte krokov do popisu* $v = x\tilde{m}\tilde{q}a_2a_3i_1j_1j_2i_2i_3\tilde{q}_u$, *kde* $\tilde{q}_u$ = *LOCATE/ a* $j_1, j_2 \in \mathbb{N}$ *sú ľubovoľné.*

Dôkaz: Predpokladáme, teda, že stroj *U*, sa nachádza v stave LOCATE@ a indexy prechodovej a stavovej hlavy sú obe rovné 2, pretože na týchto pozíciách ich ponechá stav LEFT. Stav LOCATE@ sa bude pýtať, či z prechodovej pásky načíta symbol @ a ako vidíme hneď v prvom kroku k tomu dôjde, potom posunie prechodovú hlavu doprava a prepne sa do stavu FINDSTATE, v ktorom bude porovnávať postupne znak za znakom symboly prechodovej a stavovej pásky. Z toho ako je nadefinované *m* a ako vyzerá obsah stavovej pásky vidíme, že porovnávame reprezentácie dvoch stavov. Pokiaľ nenarazíme na rozdiel a dôjdeme až k symbolu / na stavovej a prechodovej páske tak potom sú tieto stavy rovnaké a prepneme sa do stavu LOCATE/, spolu s posunom prechodovej a stavovej pásky doprava, teda tvrdenie v tomto prípade platí. To, že za popisom stavu na prechodovej páske bude symbol / a určite nie symbol @ vyplýva z toho, že *M*, v odpovedajúcom popise neukončí svoj chod a teda v *m*, musí byť definovaná pre stav *q*, aspoň jedna vstupná trojica načítaných znakov.

Ak však narazíme na rozdiel v zápise, stavy sú rôzne a my posunieme prechodovú hlavu doprava, stavovú doľava a pomocou stavu LEFTSTATE, sa presunieme na pozíciu dva na stavovej pásky, teda na začiatok zapísaného stavu a pokračujeme vykonávaním stavu LOCATE@. V tomto stave budeme postupovať doprava, až pokiaľ nenarazíme na symbol @ a následne sa znova prepneme do stavu FINDSTATE. To budeme opakovať dovtedy, kým neporovnáme dva rovnaké stavy. Vieme, že to raz nastane pretože, každý stav sa musí nachádzať v reťazci *m*, takže sme dokázali tvrdenie. ■

Samotné načítavanie vstupu do prechodovej funkcie by sme teraz radi ukončili a ukázali, že sa dostaneme do stavu WRITE, kde budeme zapisovať nové hodnoty na požadované pásky.

Tvrdenie 5 (Prečítanie vstupnej trojice symbolov):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis U je $u = x\tilde{m}\tilde{q}a_2a_3i_1j_1j_2i_2i_3q_u$, *kde* q_u = *LOCATE/,* $j_1, j_2 \in \mathbb{N}$ *sú podľa predchádzajúceho tvrdenia a* $\tilde{q} = \triangleright|q|/$, *kde q je aktuálny stav stroja M. Predpokladáme, že M v odpovedajúcom popise neukončí svoj chod. Potom sa U dostane v konečnom počte krokov do popisu* $v = x\tilde{m}\tilde{q}a_2a_3i_1\tilde{j}_1\tilde{j}_2i_2i_3\tilde{q}_u$, *kde platí, že* $\tilde{q}_u$ = *WRITE a* $\tilde{j}_1, \tilde{j}_2 \in \mathbb{N}$ *sú ľubovoľné.*

Dôkaz: Keď sa stroj dostane do stavu LOCATE/ prvýkrát, tak vieme, že na stavovej páske ukazuje na pozíciu obsahujúcu prvý prázdny symbol stroja *U* v poradí. Pričom sa táto hlava za celý čas nepohne a zostane na mieste. Index prechodovej pásky sa nachádza na prvom pamäťovom bloku za popisom aktuálneho stavu v reťazci *m*. Tam sa nachádza symbol /, to vieme, z predpokladu, že *M* neukončí v tomto momente svoj chod. Takže stav LOCATE/ sa hneď prepne do stavu READ, prechodová hlava sa posunie doprava a ešte predtým zapíšeme na stavovú pásku prázdny symbol $\square_u$. Keby to však tak nebolo a do stavu LOCATE/ by sme sa dostali na ľubovoľnej pozícií napravo od tohto prvého symbolu / a naľavo od posledného symbolu / pred symbolom @, ktorý sa v reťazci *m*, nachádza ako ďalší v poradí, tak rovnako z predpokladu, že *M* v tomto popise neukončí svoj chod, vieme, že v konečnom počte krokov, posúvaním doprava, narazíme na symbol / a pokračujeme v stave READ, spolu s už opísaným posunom a zápisom prázdneho symbolu na stavovú pásku.

V stave READ budeme zapisovať pomocné symboly na stavovú pásku a porovnávať symboly na prechodovej páske v poradí so symbolmi na vstupnej, pracovnej a výstupnej páske. Pokiaľ sa všetky rovnajú, prepneme sa do stavu WRITE a teda tvrdenie platí. Ak narazíme na nerovnosť, tak pokračujeme opäť v stave LOCATE/, až pokým nenájdeme rovnosť vo všetkých troch prípadoch. Čo, ako vieme, s určitosťou nastane, na základe predpokladu, že *M*, v tomto momente neukončí svoj chod. Takže sme dokázali požadované tvrdenie. ■

Vieme teda, že sa stroj *U*, dostane zo stavu u^U, ako ho určuje Veta 2, až k vykonávaniu výstupu odpovedajúceho kroku prechodovej funkcie stroja *M*. Rozložíme si teda krok prechodovej funkcie na tri možné prípady, ktoré keď budeme rôzne spájať dostaneme každý možný výstup tejto prechodovej funkcie.

Ako prvý si vezmime prípad, kedy by prechodová funkcia stroja *M*, zmenila len jeden symbol na ľubovoľnej z pások. Ako si uvedomíme, tak bez ujmy na všeobecnosti, môžeme predpokladať, že to bude výstupná páska. Zápis na pracovnú pásku by sme boli schopný nahliadnuť podobným tvrdením.

Tvrdenie 6 (Zápis symbolu):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis M je $u = xa_2a_3i_1i_2i_3q$, $a \in \Gamma$ *je symbol na výstupnej páske na pozícií* i_3. *Predpokladáme, že M v tomto popise neukončí svoj chod a dostane sa do popisu* $v = xa_2\hat{a}_3i_1i_2i_3q$, *kde* $\hat{a}_3$ *sa od* a_3 *líši v tom, že na pozíciu* i_3 *je zapísaný, namiesto symbolu a, symbol* $b \in \Gamma$. *Potom sa U dostane v konečnom počte krokov z odpovedajúceho popisu* u^U *do odpovedajúceho popisu* v^U.

Dôkaz: Tvrdenie 4 a Tvrdenie 5 nám umožňujú predpokladať, že *U* sa nachádza v stave WRITE. Zo stavovej pásky načítame symbol 0 a teda symbol z prechodovej pásky zapíšeme na i_2 pozíciu na pracovnej páske, ale na základe predpokladu, že v kroku *M*, ktorý simulujeme sa obsah pracovnej pásky nezmení, vieme, že tento symbol je rovnaký, ako tam bol už zapísaný a teda použitím podobného tvrdenia na pracovnú pásku dostaneme, že sa reťazec a_2 nezmení. Takže sa ocitáme v situácií, kedy z prechodovej pásky načítavame symbol *b*, z výstupnej pásky zase symbol *a*, a na stavovej páske je zapísaný symbol 1. Z definície prechodovej funkcie vidíme, že sa symbol *b* zapíše na výstupnú pásku na požadovanú pozíciu a pokiaľ sú rovnaké tak $a_3 = \hat{a}_3$, pokiaľ sú však symboly *a* a *b* rôzne, tak má $\hat{a}_3$ požadovaný tvar. Dôkazy pre Tvrdenie 7 a Tvrdenie 8 ukážu, že ak v kroku prechodovej funkcie nemeníme pozície hláv a stav *q*, tak stavy MOVE a CHANGESTATE nezmenia popis stroja *U*. Takže sa stroj dostane do požadovaného popisu a dokázali sme tvrdenie. ■

Teraz sa budeme zaoberať posunom hlavy v kroku prechodovej funkcie stroja *M*. Bez ujmy na všeobecnosti môžeme predpokladať, že budeme posúvať vstupnú hlavu doprava, zvyšné prípady by sme nahliadli obdobnými tvrdeniami.

Tvrdenie 7 (Posun hlavy):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis M je $u = xa_2a_3i_1i_2i_3q$*. Predpokladáme, že M v tomto popise neukončí svoj chod a dostane sa do popisu* $v = xa_2a_3\hat{\imath}_1i_2i_3q$*, kde* $\hat{\imath}_1 = i_1 + 1$*. Potom sa U dostane v konečnom počte krokov z odpovedajúceho popisu* u^U *do odpovedajúceho popisu* v^U*.*

Dôkaz: Tvrdenie 4 a Tvrdenie 5 spolu s dôkazom pre Tvrdenie 6 hovoria, že sa dostaneme do stavu MOVE, pričom stav WRITE nezmení obsah pracovnej a výstupnej pásky. Symbol, ktorý v tomto momente načítame z prechodovej pásky, bude rozhodovať o posune vstupnej pásky a ako vidíme bude sa jednať o symbol znázorňujúci posun vpravo a teda v popise k indexu vstupnej pásky pripočítame jednotku a tým dostaneme požadované $\hat{\imath}_1$. Následne ďalšie dva symboly budú reprezentovať zotrvanie pracovnej a výstupnej hlavy na mieste, takže sa ich indexy nezmenia. Takto máme vyriešený posun doprava a zotrvanie na mieste, pričom posun doľava, je podobný ako doprava s tým, že pokiaľ $i_1 = 1$, tak sa jedná o zotrvanie na mieste. Z dôkazu pre Tvrdenie 8 následne vyplynie, že ak nemeníme stav, tak to nemá vplyv na popis stroja *U*, a teda sa dostávame do požadovaného popisu stoja *U*. ■

Nakoniec nahliadneme, ako sa zmena aktuálneho stavu stroja *M*, odzrkadlí na popise stroja *U*.

Tvrdenie 8 (Zmena aktuáneho stavu):

Nech Tvrdenie 2 definuje U a m je reťazec reprezentujúci stroj M. Nech aktuálny popis M je $u = xa_2a_3i_1i_2i_3q$*. Predpokladáme, že M v tomto popise neukončí svoj chod a dostane sa do popisu* $v = xa_2a_3i_1i_2i_3\hat{q}$*. Potom sa U dostane v konečnom počte krokov z odpovedajúceho popisu* u^U *do odpovedajúceho popisu* v^U*.*

Dôkaz: Tvrdenie 4 a Tvrdenie 5 spolu s dôkazmi pre Tvrdenie 6 a Tvrdenie 7 hovoria, že sa dostaneme do stavu CHANGESTATE, pričom stavy WRITE a MOVE nezmenia obsah pracovnej a výstupnej pásky a nezmenia ani pozíciu ich hláv. Index prechodovej pásky bude v tomto momente odkazovať na pamäťový blok, v ktorom sa bude nachádzať symbol 0 alebo 1 a bude označovať posledný znak reťazca binárnej reprezentácie stavu $\hat{q}$. Stavová páska bude najskôr ukazovať na symbol /, ale v prvom kroku stavu CHANGESTATE to zmeníme a budeme na tejto páske, ukazovať na posledný znak reprezentácie stavu q. Postupne teda znak za znakom budeme symboly reprezentujúce $\hat{q}$ zapisovať na stavovú pásku, až kým obsah stavovej pásky nebude v tvare $\triangleright|\hat{q}|/$ teda sme zmenili odpovedajúci popis stroja U. Pričom ak sa stavy q a $\hat{q}$ rovnajú vidíme, že popis stroja U nijak nezmeníme. Stroj U sa z tohto stavu prepne do stavu LEFT a následne LOCATE@, kde ako vieme existuje požadovaný odpovedajúci stav stroja U. ■

Spoločne sa tieto tri dôkazy dopĺňajú a teda vieme, že spoločne platí Tvrdenie 6, Tvrdenie 7 a aj Tvrdenie 8. Z týchto troch tvrdení je následne jednoznačným dôsledkom aj Veta 2, ktorú sme teda takto dokázali a môžeme ju využiť pri požadovanom dôkaze pre Tvrdenie 2.

Zostáva nahliadnuť, že stroj U ukončí svoj chod práve vtedy, keď ho ukončí aj stroj M. To znamená, že ak existuje taký popis stroja M, do ktorého sa tento stroj dostane v konečnom počte krokov, že pre tento popis nemáme definovanú prechodovú funkciu stroja M. Potom sa stroj U, v konečnom počte krokov, dostane z odpovedajúceho popisu do popisu, v ktorom nemá definovanú prechodovú funkciu. Takže, dôjde pre oba stroje v rámci danej terminológie k ukončeniu ich chodu s rovnakým obsahom vstupnej, pracovnej aj výstupnej pásky. Naformulujme si to do nasledujúcej vety.

Veta 3 (Ukončenie chodu):

Nech Tvrdenie 2 definuje U, m je reťazec reprezentujúci stroj M a nech M ukončí svoj chod pre vstup x. Ak je reťazec $u = xa_2a_3i_1i_2i_3q$ *okamžitým popisom stroja M pre vstup x, v ktorom M ukončí svoj chod. Potom sa stroj U dostane, v konečnom počte krokov, z odpovedajúceho popisu* u^U *do popisu* $v^U = x\tilde{m}\tilde{q}_1a_2a_3i_1j_1j_2i_2i_3q_u$, *v ktorom ukončí svoj chod pre vstup (x, m).*

Dôkaz: V prechodovej funkcii stroja M, môžu nastať dva prípady. Ako prvé môže nastať, že pre stav $q \in Q$, nemáme v prechodovej funkcii definovanú žiadnu vstupnú trojicu. V takomto prípade bude stroj U, postupovať nasledovne. Predpokladajme, že máme na stavovej páske zapísanú bitovú reprezentáciu stavu q a hlavy prechodovej a stavovej pásky sa nachádzajú na druhej pozícií, prvej neobsahujúcej štartovací symbol. To môžeme predpokladať vďaka stavom

START, COPY a LEFT, ktoré sa vykonajú na začiatku chodu celého stroja, prípadne sa stav LEFT vykoná pred každým simulovaným krokom prechodovej funkcie stroja *M*. Zo stavu LEFT sa stroj *U*, vždy prepne do stavu LOCATE@, v ktorom bude hľadať symbol @ na prechodovej páske. Uvedomme si, že v konečnom počte krokov tento symbol nájde a prepne sa do stavu FINDSTATE. Kde začne porovnávať symboly bezprostredne za týmto symbolom, v poradí so symbolmi uloženými na stavovej páske. Keďže, môže dôjsť k tomu, že porovnané symboly sa nezhodujú vtedy prejde stroj *U*, do stavu LEFTSTATE, čím vráti stavovú hlavu späť na druhú pozíciu a pokračuje v hľadaní ďalšieho symbolu @ v poradí v stave LOCATE@. Všimnime si, že prechodová hlava sa vždy posúva v týchto stavoch len jedným smerom a prejdeme takto každý symbol na prechodovej páske, takže nevynecháme jediný symbol @. Keďže sa na požadovaných pozíciách prechodovej pásky nachádza len reťazec *m*, v ktorom ako vieme, sa musí nachádzať stav *q*, tak tento stav vždy nájdeme. V reťazci *m*, je tento stav ukončený ďalším symbolom @. Čo, ako je nadefinované v prechodovej funkcii stroja *U*, znamená ukončenie chodu tohto stroja v popise, kde q_u = FINDSTATE a indexy prechodovej a stavovej pásky môžu byť ľubovoľné.

Druhý prípad je, že stav *q* má definované nejaké trojice, ale aktuálne načítaná trojica nie je nadefinovaná. V takomto prípade ako sme si ukázali vždy nájdeme stav *q*, v reťazci *m* a keďže tento stav nie je zakončený symbolom @ ale symbolom /, tak sa prepneme do stavu LOCATE/. Vieme, že v konečnom počte krokov tento symbol nájdeme a prepneme stroj *U* do stavu READ. V tomto stave budeme porovnávať v poradí prvý symbol za nájdeným symbolom /, so symbolom načítaným zo vstupnej hlavy. Ak sa rovnajú pokračujeme na druhý symbol a pracovnú hlavu. Ak sa rovnajú, prejdeme na tretí symbol a výstupnú hlavu. Keďže však daná trojica symbolov načítaných z hláv nie je pre daný stav definovaná, tak vždy narazíme na to, že aspoň jedna z týchto troch rovností nenastane a teda sa vraciame späť do stavu LOCATE/. Kde teda dôjdeme na koniec aktuálneho stavu a narazíme v konečnom počte krokov na symbol @. V takomto prípade ukončíme chod stroja *U* v popise kde q_u = LOCATE/ a indexy prechodovej a stavovej pásky sú ľubovoľné.

Zostáva nahliadnuť, že platí aj opačná implikácia, tá však vyplýva z pozorovania, že stroj U, skončí len v týchto dvoch prebratých prípadoch. Veta 2 poskytuje hľadané pozorovanie, pretože vidíme, že v chodoch medzi jednotlivými krokmi prechodovej funkcie stroja M, musia byť všetky kroky stroja pre jeho prechodovú funkciu definované, inak by sme nedošli do požadovaného stavu z tejto vety, čo by bol spor s touto vetou. Jediné čo by mohlo nastať je ukončenie ešte v štartovacom stave START, ale ten ako sa dá ľahko uvedomiť má všetky stavy do ktorých sa dostane nadefinované. ■

V tomto momente máme všetky pomocné vety a tvrdenia dokázané. Spoločne Veta 1, Veta 2, Veta 3 a Tvrdenie 3 ako priamy dôsledok hovoria o tom, že náš stroj U, je univerzálny Turingov stroj, pričom Definícia 4 na str. 9 podáva formálnu definíciu tohto pojmu. Bod (a) z tejto definície je splnený automaticky. Veta 3 nám hovorí že stroj U, ukončí svoj chod práve vtedy keď stroj M. Veta 1, Veta 2 a Tvrdenie 3 následne hovoria, že obsah vstupnej, pracovnej a výstupnej pásky stroja U je v každom okamihu jeho chodu zhodný s odpovedajúcim okamihom pre stroj M. Takže sa rovnajú aj výstupy oboch strojov a požadovaná inklúzia $V \subseteq V_u$ platí pre každý stroj M triviálne, podľa toho, ako sme si ho nadefinovali. Takže v definícií platí aj bod (b). Čím sme dokázali univerzálnosť stroja U. ■

Záver

Na záver práce by som rád uviedol pár slov k jej výsledkom a poznatkom získaným počas jej písania. Výsledok tejto práce nie je tvorený len samotným zistením, že existuje nejaký univerzálny Turingov stroj. Výsledok práce spočíva hlavne v zavedení vlastného príkladu Turingovho stroja a v dokázaní jeho univerzálnosti. Práca teda umožňuje prípadným čitateľom, pochopiť existenciu univerzálnej výpočtovej metódy na konkrétnom príklade. V práci sa zameriavam na vysvetlenie jednotlivých pojmov v danej problematike. Turingov stroj, ktorý som v tejto práci zaviedol nemusí byť najefektívnejší a samozrejme sa nemusí jednať o jeho najjednoduchšiu verziu. Je naozaj možné, mať univerzálny stroj pracujúci s menším počtom pások, nad menšou abecedou, prípadne s menšou množinou stavov. To však nebolo zámerom tejto práce. Snažil som sa podať taký náhľad na danú teóriu, ako som aj viackrát v práci zdôrazňoval, aby bolo jasne viditeľné, čo sa za danými pojmami deje a bol naozaj každý schopný zostrojiť a zadefinovať takýto univerzálny Turingov stroj.

Pre dokončenie práce som pracoval s rôznymi literárnymi a internetovými zdrojmi, ktoré mi poskytli teoretický základ k danej teórií. Dôležitým krokom však bolo praktické porozumenie daným pojmom. Využil som teda znalosti nadobudnuté počas štúdia, nato aby som si dokázal zrekonštruovať niekoľko príkladov Turingových strojov aj so zložitejšími definíciami pomocou programovacích jazykov. Na základe týchto praktických príkladov, som následne získal lepšiu orientáciu v daných pojmoch. A využil to hlavne pri vytváraní vhodnej reprezentácie Turingových strojov, ktorej správna voľba sa ukázala ako kľúčová pre celú problematiku.

Ako som si v týchto príkladoch uvedomil, pojem Turingov stroj môžeme zjednodušene interpretovať ako formálny popis algoritmu, ktorý dokážeme využiť napríklad aj na návrh programovacieho jazyka. Toto zistenie mi prinieslo dostatočné pochopenie, ako jednotlivé stroje fungujú nato, aby som bol schopný vytvoriť popis potenciálne univerzálneho Turingovho stroja. A následne, túto intuíciu podporiť aj formálnym dôkazom, s využitím schopností nadobudnutých počas štúdia.

Zoznam použitej literatúry

ARORA, Sanjeev. a BARAK, Boaz. *Computational complexity: a modern approach*. Cambridge: Cambridge University Press, 2009, s. 7-37. ISBN 978-0-521-42426-4.

COOPER, S. Barry. a LEEUWEN, Jan Van. *Alan Turing: His Work and Impact*. Waltham: Elsevier Science, 2012, s. 13-115. ISBN: 978-0-12-386980-7.

SIPSER, Michael. a LEE, Marie. *Introduction to the theory of computation*. 3rd ed. Boston: Course Technology Cengage Learning, 2013, s. 165-192. ISBN 978-1-133-18779-0.

TURING, Alan Mathison. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society* [online]. 42. 1937, s. 230-265 [cit. 2019-03-14]. Dostupné z: https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf.

VACEK, Jan. *Universální Turingův stroj* [online]. Praha, 2010 [cit. 2019-03-16]. Dostupné z: https://is.cuni.cz/webapps/zzp/detail/78571. Bakalářská práce. Matematicko- fyzikální fakulta, Univerzita Karlova. Vedoucí práce Jan Krajíček.

Zoznam tabuliek